

**UNIVERSIDAD CARLOS III DE MADRID**

**ESCUELA POLITÉCNICA**

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

**TRABAJO FIN DE GRADO**

***SIMULACIÓN DE LA AUTOMATIZACIÓN  
DE PROCESOS EN UNITY PRO:  
VIVIENDA DOMÓTICA***

**AUTOR:** ALBERTO HERRÁEZ PULIDO

**TURORA:** MARÍA DOLORES BLANCO ROJAS

**CURSO:** 2012-2013

---

*Quisiera agradecer el apoyo incondicional de mi familia, en concreto, de mis padres y mi hermano, gracias a vosotros he conseguido cumplir uno de los sueños que quería lograr a lo largo de mi vida, sé que sin vuestra ayuda y todo el cariño que me demostráis me hubiera sido imposible, os quiero mucho.*

---

*También, a mi pareja, sin la cual, no habría llegado hasta aquí, quiero darte las gracias por ayudarme en los malos momentos, por apoyarme y creer en mí de manera incondicional, te quiero.*

---

*Por último, me gustaría agradecer los sabios consejos y aclaraciones de mi tutora, los cuales, han hecho que pueda exprimir al máximo la realización de este trabajo.*

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

*ÍNDICE DE CONTENIDOS*



## Índice de contenidos

### **1. Motivación**

1.1.	Motivación del proyecto.....	11
1.2.	Objetivo del proyecto.....	11
1.3.	Resumen estructurado del proyecto.....	12

### **2. Introducción**

2.1.	La automatización y su evolución histórica.....	14
2.2.	La domótica dentro de la automatización.....	15
2.3.	Concepto de simulador, ejemplos de uso y ventajas.....	16
2.4.	Simuladores docentes.....	17
2.5.	Sistema Scada vs simulador.....	19

### **3. Software utilizado: Unity Pro XI – Schneider**

3.1.	Descripción y características.....	21
3.2.	Lenguajes y entorno de programación.....	22
3.3.	Tablas y entorno de simulación.....	27

### **4. Diseño y solución del problema propuesto**

4.1.	Presentación del problema.....	30
4.2.	Diseño de la solución.....	30
4.2.1.	Estructura general, funcionamiento y planificación del diseño propuesto.....	30
4.2.2.	Diseño y funcionamiento de los subsistemas.....	33
4.2.2.1.	Clave de seguridad.....	33
4.2.2.2.	Iluminación.....	34
4.2.2.3.	Persianas.....	37
4.2.2.4.	Detección de gas y humo.....	40
4.2.2.5.	Detección de presencia.....	41
4.3.	Programación en Unity Pro XI.....	42
4.3.1.	Configuración de Unity Pro XI.....	42
4.3.2.	Definición de las variables necesarias.....	44
4.3.3.	Programación de la solución en SFC.....	47
4.3.3.1.	SFC principal.....	48

4.3.3.2. Macroetapas del modo fuera de casa.....	54
4.3.3.2.1. Apagar_luces.....	54
4.3.3.2.2. Bajar_persianas_ext.....	55
4.3.3.2.3. Det_presencia.....	56
4.3.3.2.4. Det_gas_humo_ext.....	56
4.3.3.3. Macroetapas del modo dentro de casa.....	57
4.3.3.3.1. Det_gas_humo.....	57
4.3.3.3.2. Up_down_persianas.....	57
4.3.3.3.3. Iluminación.....	58
4.4. Creación de tablas de variables.....	59
4.5. Diseño de la pantalla de operador.....	61
4.5.1. Funcionamiento de la pantalla de operador.....	62
4.5.2. Elementos usados en la pantalla de operador.....	65
<b>5. Simulación del diseño</b>	
5.1. Simulación utilizando tablas.....	74
5.2. Simulación utilizando pantalla de operador.....	78
5.3. Simulación conectando la pantalla de operador al autómatas....	80
5.4. Comparación entre simulaciones.....	83
<b>6. Conclusiones</b>	
6.1. Conclusiones generales.....	85
6.2. Posibles mejoras.....	86
<b>7. Anexos</b>	
7.1. Anexo I: programación de etapas y transiciones en lenguaje Ladder (LD).....	88
<b>8. Bibliografía</b>	

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

*ÍNDICE DE FIGURAS Y TABLAS*



Figura 1: Pato de Vaucanson.....	14
Figura 2: Esquema de funcionamiento de un autómata programable.....	15
Figura 3: Requisitos mínimos hardware y software Unity Pro XI.....	21
Figura 4: Ejemplo de programación IL.....	22
Figura 5: Ejemplo de programación ST.....	23
Figura 6: Ejemplo de programación FBD.....	23
Figura 7: Ejemplo de programación LD.....	24
Figura 8: Ejemplo de programación SFC.....	25
Figura 9: Creación de una sección de programación.....	26
Figura 10: Entorno de programación Unity Pro XI.....	27
Figura 11: Creación de tabla de animación.....	28
Figura 12: Creación de pantalla de operador.....	28
Figura 13: Diagrama de flujo del sistema principal.....	31
Tabla 1: Número mínimo de entradas y salidas necesarias.....	33
Figura 14: Diagrama de flujo de sistema clave de seguridad.....	34
Figura 15: Diagrama de flujo de sistema de iluminación (fuera de casa).....	35
Figura 16: Diagrama de flujo de sistema de iluminación (dentro de casa).....	36
Figura 17: Diagrama de flujo de sistema de persianas (fuera de casa).....	38
Figura 18: Diagrama de flujo de sistema de persianas (dentro de casa).....	39
Figura 19: Diagrama de flujo de sistema de detección de gas y humo.....	40
Figura 20: Diagrama de flujo de sistema de detección de presencia.....	41
Figura 21: Autómata Premiun TSX del laboratorio.....	42
Figura 22: Creación nuevo proyecto en Unity Pro XI.....	42
Figura 23: Configuración hardware del autómata.....	43
Figura 24: Creación de variables.....	44
Figura 25: Variables elementales utilizadas en el sistema.....	45
Figura 26: Variables de entrada del sistema.....	45
Figura 27: Variables de salida del sistema.....	46
Figura 28: Nomenclatura de la dirección de las variables.....	46
Figura 29: Comprobación e introducción de la dirección de las variables.....	47
Figura 30: SFC principal.....	49
Figura 31: SFC clave seguridad.....	50
Figura 32: Introducción de tiempo de supervisión.....	51

Figura 33: Detalle SFC principal 1.....	52
Figura 34: Detalle SFC principal 2.....	52
Tabla 2: Código binario de conmutación entre modos (*1->botón pulsado; 0-> sin pulsar).....	52
Figura 35: Detalle conjunción alternativa SFC principal.....	53
Figura 36: Detalle camino fuera_casa SFC principal.....	53
Figura 37: Detalle camino dentro_casa SFC principal.....	54
Figura 38: Macroetapa apagar_luces modo fuera de casa.....	55
Figura 39: Macroetapa bajar_persianas modo fuera de casa.....	55
Figura 40: Macroetapa det_presencia modo fuera de casa.....	56
Figura 41: Macroetapa det_gas_y_humo modo fuera de casa.....	57
Figura 42: Macroetapa up_down_persianas modo dentro de casa.....	58
Figura 43: Macroetapa iluminación modo dentro de casa.....	58
Figura 44: Tablas de animación.....	60
Figura 45: Tabla de animación “entradas” .....	60
Figura 46: Tabla de animación “salidas” .....	61
Figura 47: Pantalla principal del simulador domótico.....	62
Figura 48: Pantalla de ayuda del simulador domótico.....	63
Figura 49: Barra de herramientas IOSEditor.....	65
Figura 50: Ejemplo de configuración de forma.....	66
Figura 51: Ejemplo de configuración de texto.....	67
Figura 52: Ejemplo de configuración de imagen.....	68
Figura 53: Ejemplo de configuración de botón examinar pantalla.....	69
Figura 54: Ejemplo de configuración de botón de comando.....	69
Figura 55: Elementos gráficos incorporados en el simulador domótico.....	70
Figura 56: Librería de pantallas de operador Unity Pro XI.....	70
Figura 57: Barra de herramientas Api.....	73
Figura 58: Sistema inactivo.....	74
Figura 59: Sistema activo (sin botones).....	74
Figura 60: Sistema activo (con botones).....	74
Figura 61: Alarma bloqueo clave de entrada.....	75
Figura 62: Sistema activado (con clave introducida).....	75
Figura 63: Modo manual activado.....	76
Figura 64: Modo fuera de casa activado.....	76



Figura 65: Modo dentro de casa activado.....	77
Figura 66: Modo dentro de casa activado.....	78
Figura 67: Gas/humo on.....	78
Figura 68: Alarma gas/humo.....	78
Figura 69: iluminación on.....	78
Figura 70: Iluminación activada.....	79
Figura 71: Apagado iluminación sin presencia.....	79
Figura 72: Iluminación selectiva.....	79
Figura 73: Telefast.....	80
Tabla 3: Configuración 1 y 2 módulo hardware.....	81
Tabla 4: Configuración 3 módulo hardware.....	81
Figura 74: Modo manual fuera de casa activado.....	82
Figura 75: Bajada persianas y luces.....	82
Figura 76: Alarma error bajada de persianas.....	82
Figura 77: Alarma robo.....	82

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 1: MOTIVACIÓN***



## 1.1. Motivación del Proyecto

En el presente proyecto, estudiaremos el software Unity Pro XI v6.o utilizado en los autómatas TSX Premium de Schneider del laboratorio de Automatización del departamento de Ingeniería de Sistemas y Automática. Anteriormente, se trabajaba en los laboratorios con el software PL7 Junior, pero se ha observado que Unity Pro tiene múltiples ventajas frente a su antecesor. Una de las más significativas es la posibilidad de realizar “Pantallas de Operador”, estas pantallas son simuladores gráficos de los procesos programados en los autómatas por los alumnos, de esta forma se puede observar gráficamente el desarrollo del proceso sin necesidad de utilizar el autómata, con las ventajas que ello conlleva. Este proyecto fin de carrera se centra en realizar un simulador gráfico para solucionar un determinado problema propuesto.

## 1.2. Objetivo del Proyecto

El objetivo del proyecto es crear un entorno de simulación para la resolución de problemas de automatización utilizando las herramientas proporcionadas por Unity Pro XI v6.o., debiendo realizar:

- La solución del problema propuesto, implementada y programada en UnityPro v6.o., ya sea en lenguaje SFC, LD, ST, IL o FBD (o la combinación de varios).
- Crear una tabla de animación que permita simular el proceso propuesto mediante el forzado de los valores de las variables.
- Crear una pantalla de operador que permita simular el proceso propuesto mediante la activación de los botones de la pantalla.
- Conectar la pantalla del operador con los módulos de E/S del autómata disponible en el laboratorio, y comprobar que el sistema se comporta correctamente.

Además de esto, el simulador (pantalla de operador) debe cumplir algunos requisitos básicos como, por ejemplo:

- Debe ser sencillo e intuitivo, con el fin de facilitar el uso a la persona que realiza la simulación.
- Debe mostrar los cambios de estado de variables y los sucesos del proceso de manera clara.
- En la medida de lo posible, contendrá una leyenda explicativa de los diferentes iconos, botones, señales de aviso, etc...
- Imitar de una manera más precisa o no (dependiendo el uso del simulador y según sea necesario) el proceso que representa dicho simulador.

### 1.3. Resumen Estructurado del Proyecto

El proyecto ha sido estructurado en 8 capítulos incluyendo la bibliografía y los anexos (ambos excluidos en este resumen), en los cuales, se expresa cómo ha sido realizado paso a paso el presente proyecto. A modo de resumen, en este apartado veremos una corta descripción de los temas tratados en cada capítulo.

- Capítulo 1: Se explica la motivación que lleva a plantear este proyecto, así como los objetivos previos a cumplir y un resumen estructural del proyecto.
- Capítulo 2: El objetivo es presentar los conceptos clave necesarios para introducirnos en la situación del proyecto, comienza definiendo qué es un autómatas y realizando un breve repaso histórico de su evolución. Más tarde, se hablará brevemente de la domótica, de sus ventajas, su implantación en la actualidad y sus áreas de aplicación. Para terminar, hablaremos de los simuladores, de sus ejemplos de uso, sus ventajas, los compararemos con los sistemas Scada y entraremos un poco más en detalle en simuladores docentes.
- Capítulo 3: Aquí se hablará de manera breve del software utilizado, describiendo sus características generales y sus múltiples ventajas. También, se realizará una explicación del entorno de programación y los lenguajes de programación disponibles; para terminar, se explicará cómo funciona el entorno de simulación (tablas y pantallas de operador).
- Capítulo 4: Se presenta el problema propuesto para a continuación explicar con todo detalle (modelado del sistema mediante SFC, SFC programados, justificación de la solución, explicación del funcionamiento) la solución sugerida, tanto de manera general (SFC general), como cada subsistema de manera independiente. En este capítulo, también se explicará la configuración realizada al autómatas, las variables utilizadas, la programación de la solución y la creación de las tablas de variables y pantallas de operador.
- Capítulo 5: Se explica con todo detalle cómo se han realizado las simulaciones, diferenciando entre las realizadas mediante tabla de variables, pantalla de operador y pantalla de operador conectada al autómatas. Se compararán los resultados obtenidos en cada simulación y se analizarán las ventajas e inconvenientes que presentan cada una de ellas.
- Capítulo 6: Para terminar, en el capítulo 6 se hace una visión general del resultado del proyecto, explicando los objetivos cumplidos, las conclusiones sacadas de la realización de dicho proyecto y, finalmente, una opinión personal expresando posibles mejoras y la evolución de esta clase de sistemas.

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

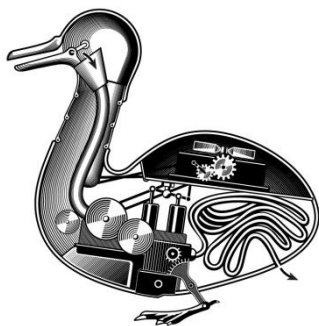
***CAPITULO 2: INTRODUCCIÓN***



## 2.1. La automatización y su evolución histórica

Lo primero que debemos conocer es a qué llamamos automática, según la RAE, la automática es “La ciencia que trata de sustituir en un proceso el operador humano por dispositivos mecánicos o electrónicos”. De igual manera, nos interesa saber el significado del término autómeta, “Máquina que imita la figura y los movimientos de un ser animado”. [1]

Una vez que sabemos lo que significa automática y autómeta, vamos a hacer un pequeño resumen a lo largo de la historia para ver cómo han evolucionado estos sistemas. Los primeros ejemplos se trataban de muñecos o sistemas sencillos, que realizaban movimientos o acciones de manera automática, por ejemplo, en el año 1500 a.C en Etiopía se construyó una estatua que emitía sonidos al ser iluminada por los rayos del sol; entre el 400-397 a.C en Grecia otra máquina simulaba el vuelo; ya en la Edad Media se siguió con esta tendencia, un ejemplo claro es el Gallo de Estrasburgo, el cual, estaba situado en el reloj de la catedral y al dar las horas movía el pico y la cola; uno de los ejemplos más significativos data del siglo XVIII con el pato mecánico de Vaucanson (ver



**Figura 1: Pato de Vaucanson**

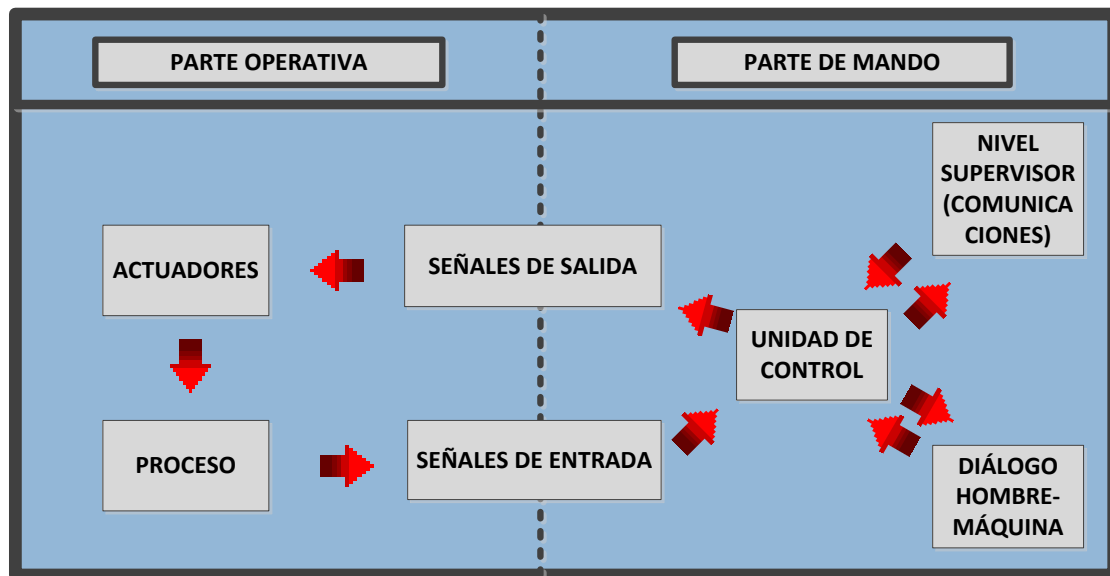
figura 1), dicho pato podía comer de la mano y lo más impresionante, digerir esta comida y evacuarla por un ano artificial. No obstante, no es hasta principios del siglo XIX cuando se empezaron a incorporar algunos automatismos en la industria textil, como las hiladoras textiles o telares mecánicos; más tarde, se comienzan a incorporar en las industrias mineras y metalúrgicas, pero el impulso realmente importante surgió en la industria del automóvil en 1960, cuando General Motors quiso mejorar los sistemas de control eléctricos de relés que tenían incorporados en sus fábricas de vehículos, pretendían que los controladores tuvieran flexibilidad y

sencillez de programación, larga vida útil y uso en ambientes adversos, aquí fue cuando se introdujo el primer autómeta programable. [2]

Un autómeta programable es un equipo electrónico de control que realiza un encadenamiento automático y continuo de operaciones, siendo gobernado por un programa de control previamente almacenado en su memoria, son capaces de procesar información de entrada y establecer las salidas correspondientes.

En la actualidad, los autómetas incorporan dos partes (ver figura 2), una parte operativa formada por los accionadores, los cuales, interaccionan con las máquinas (proceso) y una parte de mando que se encarga de controlar el sistema mediante el programa contenido en la unidad de control, esta última, ejecuta las señales de salida necesarias para activar los accionadores (preaccionadores), controla el proceso mediante la información que recibe de los sensores (captadores), se comunica (mediante profibus, redes de comunicaciones, internet etc...) con otras unidades para sincronizarse y enviar

información para supervisión y aparte, contiene un sistema para que el usuario observe el estado del proceso y pueda intervenir de manera activa (diálogo hombre-máquina). [2]



**Figura 2: Esquema de funcionamiento de un autómata programable**

Uno de los puntos fuertes de los autómatas programables es la estandarización de su hardware, esto les hace ser uno de los sistemas de control más usados actualmente en la industria para controlar el funcionamiento de máquinas, plantas o procesos industriales de todo tipo y envergadura. Entre sus múltiples ventajas, destacan la flexibilidad y facilidad de programación, el ahorro de tiempo en la elaboración de proyectos (debido a su facilidad de modificación), su tamaño reducido, el bajo nivel de mantenimiento, la resistencia al ruido eléctrico, a impactos, etc... . Gracias a todas estas ventajas es necesaria su utilización tanto en entornos industriales como no industriales. [3]

## 2.2. La domótica dentro de la automatización

Desde principios de los años 90 hasta la actualidad hemos podido observar como la automatización ha ido expandiendo su campo de actuación a otros sectores diferentes al industrial. Hoy en día, hay prototipos de vehículos inteligentes que son capaces de conducir con cierta autonomía (coche de Google), trenes que no llevan conductor, aviones no tripulados (UAS), viviendas inteligentes (domótica) e infinitas aplicaciones más.

Centrándonos en la domótica, por vivienda inteligente entendemos a aquella que permite una mayor calidad de vida a través de la tecnología, ofreciendo un aumento del bienestar, de la seguridad y una disminución del consumo energético. [4]

La incorporación de la automatización a las viviendas tiene múltiples beneficios como, por ejemplo, el ahorro de energía, el aumento del confort y de la calidad de vida, la seguridad, la optimización de la red de comunicaciones (acceso a las instalaciones mediante control remoto). Esto ha hecho que en los últimos años, las casas de nueva construcción lleven incorporados algunos de estos sistemas según las necesidades de los clientes. Las características más importantes en un sistema domótico son su facilidad de uso, su flexibilidad y la interconexión entre los distintos sistemas.

Existen varias áreas de aplicación: seguridad (alarmas de intrusión, detección gas /humo, control acceso...), confortabilidad (persianas, iluminación, riego automático, control climatización...), gestión de la energía (gestión de la climatización, iluminación...) y gestión de las comunicaciones (telecontrol vía internet, vía telefónica, transmisión alarmas, control electrodomésticos...). Según un estudio realizado en 2011 por la CEDOM (asociación española de domótica), en España un 23% de los usuarios centra su sistema domótico en confort, seguido por un 20% en ahorro energético y un 16% en telecomunicaciones. [5]

Cabe destacar que el uso de este tipo de viviendas irá en aumento a corto-medio plazo, debido a sus múltiples ventajas y al uso de la energía de manera más eficiente y sostenible con el medio ambiente.

### 2.3. Concepto de simulador, ejemplos de uso y ventajas

Un simulador es un aparato, por lo general informático, que permite la reproducción de un sistema, reproduce tanto las sensaciones físicas (velocidad, aceleración, percepción del entorno) como el comportamiento de los equipos de la máquina simulada, a su vez, permite a la persona practicar en el manejo del sistema sin necesidad de utilizar dicha máquina, preparándola ante todo tipo de situaciones, con el objetivo de saber afrontarlas si se le plantean en la vida real. [6]

Los simuladores son ampliamente usados en muchos ámbitos por sus innumerables ventajas, estos van desde ámbitos profesionales como, por ejemplo, los pilotos de avión, hasta usos de ocio como videojuegos. En España, los simuladores tienen un alto grado de implantación destacando sectores como: [6]

- Conducción: Existen simuladores usados en autoescuelas por los alumnos para prever situaciones con anterioridad a clases prácticas y de esta manera estar prevenidos.
- Aviación: Los pilotos utilizan simuladores de vuelo antes de ponerse al mando de los aviones reales, también existen simuladores para probar nuevos aviones antes de su primer vuelo.
- Trenes: Permiten controlar trenes.
- Negocio: Permiten simular entornos empresariales.



- Clínico médico: Permiten realizar simulaciones de diagnósticos clínicos de pacientes.
- Musicales: Permiten simular instrumentos musicales.
- Ocio: Permiten controlar todo tipo de situaciones, como construcción de casas, la vida de una persona (Sims), simuladores de guerra y múltiples ejemplos más.

Como hemos podido observar, los simuladores están introducidos en sectores ingenieriles, empresariales, medicinales, de ocio... ,y esto se debe a las múltiples ventajas que aportan este tipo de sistemas, algunas de ellas son:

- Permiten la aplicación de conocimiento de forma activa por parte del alumno a la solución de problemas.
- Mejoran la transferencia de conocimiento al plantearse situaciones simuladas de la realidad.
- Aumentan la comprensión de conceptos abstractos, que vistos gráficamente son más sencillos de comprender.
- Aumentan la motivación de los alumnos.
- Preparan ante situaciones peligrosas o inesperadas.
- Aportan experiencia en uso de máquinas de difícil manejo, tales como aviones, robots quirúrgicos, trenes, etc...
- Permiten simplificar procesos, con el objetivo de simular partes específicas de estos.
- Pueden ser utilizados para prever resultados y dimensionar proyectos antes de ser realizados, por ejemplo, estimar el rendimiento de un parque eólico.

## 2.4. Simuladores docentes

Una vez comentada la idea general de los simuladores, su implantación en todos los ámbitos y sus ventajas e inconvenientes más destacados, explicaremos con mayor detalle los simuladores docentes, debido a que este proyecto ha sido desarrollado en el marco de los simuladores docentes.

Los simuladores docentes hacen que los alumnos puedan realizar prácticas muy semejantes a la realidad y ampliar su grado de conocimiento en determinados temas de estudio. Estos simuladores pueden contar con varios métodos de uso en ámbitos docentes, los más destacables son:

- Utilización por parte del profesor en el aula, con el objetivo de afianzar conceptos teóricos complejos mediante un método gráfico que facilite su entendimiento a los alumnos.
- Utilización por parte del alumno, de manera que el alumno por sí mismo tenga que entender cómo funciona el proceso, por ejemplo, ver el estado de las variables del sistema representado, simular determinados ejercicios para conseguir objetivos u determinadas acciones del sistema, etc...

- Utilización guiada por parte del profesor, esta última, trata de aunar lo mejor de los dos métodos anteriores, intentando obtener el máximo provecho al interés por aprender del alumno y al conocimiento del profesor, de manera que se pueda dar libertad al alumno, pero este también pueda preguntar cuestiones al docente y ser corregido si fuese necesario.

Además, los simuladores docentes pueden representar una optimización de los procesos, principalmente enfocada a simplificar un sistema complejo y aplicar los conocimientos que sean de interés y obviar los demás. Los simuladores docentes usan el aprendizaje de tipo experimental, de manera que el alumno pueda aprender conocimientos por sí mismo y a la par, controlar mediante un entorno simulado situaciones en el mundo real, preparándolo para la toma de decisiones en su vida profesional.

Hoy en día, los simuladores son ampliamente utilizados en docencia debido a sus múltiples ventajas frente a los sistemas de aprendizaje tradicional, algunas de ellas son:

- Conllevan un ciclo de aprendizaje: reflexionar sobre los problemas, elegir una estrategia, tomar decisiones y llevarlas a cabo y, por último, ver las consecuencias y actuar ante ellas. [7] [8]
- Potencian el factor motivacional, haciendo para los alumnos el aprendizaje más sencillo, e incluso, en ocasiones puede llegar a ser divertido, hecho que mejora más aún el aprendizaje de conceptos.
- Cuentan con autoaprendizaje por parte del alumno, dándole la oportunidad de autocorregirse, ya que es él mismo el que se percata de los errores y trata de subsanarlos, reconociendo los fallos y entendiendo los problemas derivados de dicho error.
- Crean ambientes participativos, haciendo que la colaboración entre grupos para la resolución de problemas sea más eficaz.
- Permiten estimular la creatividad del alumno para proponer diferentes soluciones a problemas planteados.
- El grado de dificultad puede ir en aumento, desde un nivel sencillo con control de pocas variables del sistema, hasta llegar a un nivel de realidad muy alto, donde se controle el estado de muchas variables.

Los principales problemas asociados al uso de simuladores radican en la limitación de recursos materiales, tiempo y espacios apropiados, principalmente debido a la masificación de la educación y a los problemas presupuestarios. Normalmente, el método de enseñanza empleado es la supervisión por parte del profesor, sin embargo, el mayor inconveniente es la falta de recursos suficientes en las aulas para poder llevar a cabo este tipo de métodos de enseñanza prácticos, debido todo ello al alto coste de los equipos necesarios, si bien, es cierto que con el mayor desarrollo de software se está facilitando la implantación, pero a día de hoy todavía sigue siendo costosa.

## 2.5. Sistema scada vs simulador

Un sistema Scada “Supervisory Control And Data Acquisition (Supervisión, Control y Adquisición de Datos)”, es un software para ordenadores que permite controlar y supervisar procesos industriales a distancia, facilitando la realimentación en tiempo real con los dispositivos de campo (sensores y actuadores) y controlando el proceso automáticamente. Además, provee a diferentes sistemas y a sí mismo de toda la información que se genera en el proceso productivo (usada para supervisión, control calidad, control de producción, almacenamiento de datos, etc.) y permite su gestión e intervención. Un sistema SCADA incluye un hardware de señal de entrada y salida, controladores, interfaz hombre-máquina (HMI), redes, comunicaciones, base de datos y software. [9]

A diferencia de los simuladores, los sistemas SCADA se encargan de monitorizar procesos en tiempo real, esto hace que sean ampliamente utilizados para detectar errores de proceso y subsanarlos. Sin embargo, no tienen el control directo sobre todas las variables del sistema (los simuladores tampoco tienen total acceso a las variables reales del sistema, cuentan con una visualización de su estado), al sistema Scada llega toda la información de los demás equipos de campo, pero en la mayoría de los casos su actuación está limitada a cierto tipo de maniobras, por ejemplo, iniciar o parar el sistema, lanzar alarmas de averías, pero no tiene control sobre un cierto actuador como, por ejemplo, una válvula, sólo tiene información del estado de dicha válvula.

Otra diferencia importante entre ambos sistemas es la posibilidad para almacenar información y crear bases de datos en tiempo real de los sistemas Scada, según los datos van llegando al equipo son enviados a la base de datos, con el fin de tener un historial de la información del proceso. Esta base de datos, puede tener múltiples usos como, por ejemplo, ser utilizada para vigilar tiempos de procesos, análisis de la eficiencia del proceso para mejorar la productividad y muchos otros.

Por último, esclarecer que la mayor diferencia entre un sistema Scada y un simulador como hemos expresado en párrafos anteriores, radica en que el Scada trabaja en tiempo real (con realimentación bidireccional) y no permite el control total de las variables del sistema, sin embargo, el simulador sirve para simular procesos o partes de ellos, teniendo un mayor control sobre las variables del sistema y pudiendo observar su comportamiento simulado.

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 3: SOTFWARE UTILIZADO: UNITY PRO XL-  
SCHNEIDER***



En este capítulo se hablará brevemente del software utilizado, el Unity Pro XI V6.0, se expondrán sus características principales y se dará una visión genérica de los entornos de programación y de simulación que ofrece dicho software. [10]

### 3.1. Descripción y características

El Unity Pro XI V6.0 es un completo software multitarea que permite configurar, programar y depurar programas, es compatible con los modelos de PLC Modicon M340 PAC, Premium, Atrium y Quantum de la marca Schneider. Entre sus características y ventajas más importantes destacan:

- Permite programar en 5 lenguajes de programación diferentes: SFC, LD, FBD, IL, ST, regulados por la norma IEC 61131-3.
- Contiene un editor de programación (LL 984).
- Incorpora una librería personalizable de bloques FBD que se puede utilizar en todos los lenguajes menos en SFC.
- Posibilidad de integración de bus de campo usando la tecnología FDT/DTM.
- Tiene un editor de documentación de usuario.
- Contiene una herramienta que convierte archivos de Concept<sup>™</sup> y PL7<sup>™</sup> para que sean compatibles con el programa Unity Pro XI.
- Permite realizar pantallas de operador, incluyendo una base de imágenes.
- Se puede implementar un sistema Scada disponiendo de la red Ethernet necesaria configurada y la herramienta “Vijeo Citect”. [11] [12]

Los requisitos de hardware para que el programa pueda correr correctamente son los que vemos en la siguiente figura, la cual, aparece en el manual de Unity Pro XI. [13]

	Mínimo	Recomendado
Sistema	Pentium 800 MHz o más	1,2 GHz
Memoria RAM	256 MB	512 MB
Disco duro	2 GB	4 GB
Sistema operativo	Windows 2000 o Windows XP edición profesional	
Lector	Lector CD-ROM	Reproductor CD-ROM
Pantalla	SVGA o una pantalla de mayor resolución	
Periféricos	Ratón, teclado o un sistema de puntería	
Acceso a Internet	La solución recomendada para registrarse es Internet	

Figura 3: Requisitos mínimos hardware y software Unity Pro XI

Algunas de las características destacadas anteriormente, unidas a la normalización de los lenguajes de programación usados, hacen de este software una herramienta sencilla e intuitiva para su uso en ambientes industriales, empresariales y docentes, además, la sencillez de las interfaces de programación y simulación facilita el trabajo, reduciendo la cantidad de horas dedicadas al diseño, simulación y prueba de los sistemas desarrollados.

### 3.2. Lenguajes y entorno de programación

Como se ha comentado anteriormente, Unity Pro permite realizar la programación de las aplicaciones mediante cinco lenguajes diferentes, dos de ellos son de tipo literal (lenguaje literal estructurado o “ST” y lista de instrucciones o “IL”) y los otros tres de tipo gráfico (diagrama de contactos “LD”, diagrama de bloques funcionales “FBD” y diagrama secuencial “SFC”).

Comenzando por los lenguajes literales tenemos el lenguaje IL, compuesto por una secuencia de instrucciones ejecutadas fila a fila de arriba a abajo, cada una utiliza un operador y si son necesarios, modificadores y operandos. Este lenguaje está basado en un acumulador, por ello, un programa en IL debe comenzar con el operador “LD”, el cual, carga el valor al acumulador, aparte existen diferentes tipos de comandos y operadores para realizar cualquier función. Podemos observar un ejemplo en la figura 4.

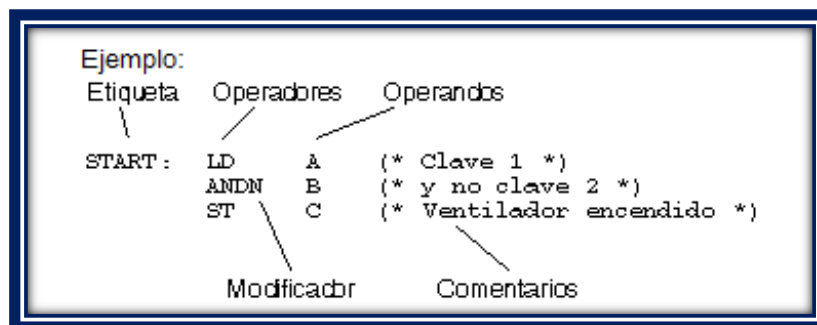


Figura 4: Ejemplo de programación IL

Más parecido a los lenguajes de programación informáticos de alto nivel es el lenguaje literal estructurado, con este lenguaje tenemos la posibilidad de realizar expresiones más complejas y anidaciones con sencillez. Para la realización de las expresiones, se usan operadores y operandos que devuelven un valor a una instrucción durante la ejecución del programa, al igual que ocurre en el caso del lenguaje IL se ejecuta fila a fila de arriba a abajo, de nuevo volvemos a ver un ejemplo en la figura 5.

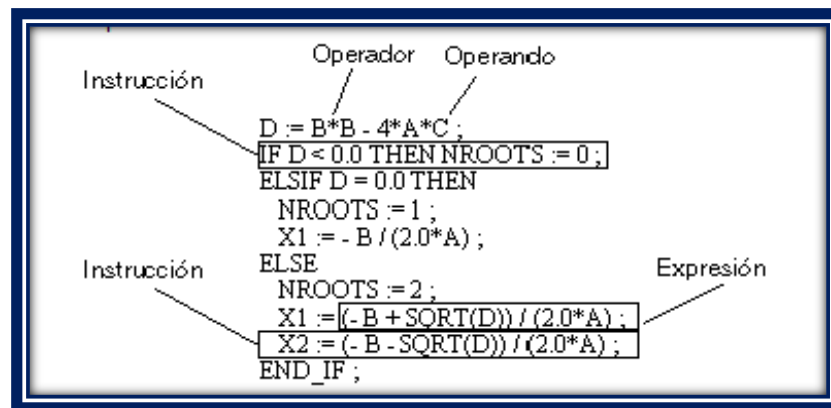


Figura 5: Ejemplo de programación ST

El FBD consiste en programar utilizando bloques de funciones, el conjunto de todos ellos se denomina FFB y cuenta con funciones elementales (EF), bloques de funciones derivados (DFB), bloques de funciones elementales (EFB), procedimientos y elementos de conexión y control, todos son bloques están programados internamente y tienen una determinada función, por ejemplo, un bloque AND tiene en cuenta que sus dos o más señales de entrada estén activas para activar la salida, en la figura 6, vemos un ejemplo aplicado a un motor de una cinta transportadora.

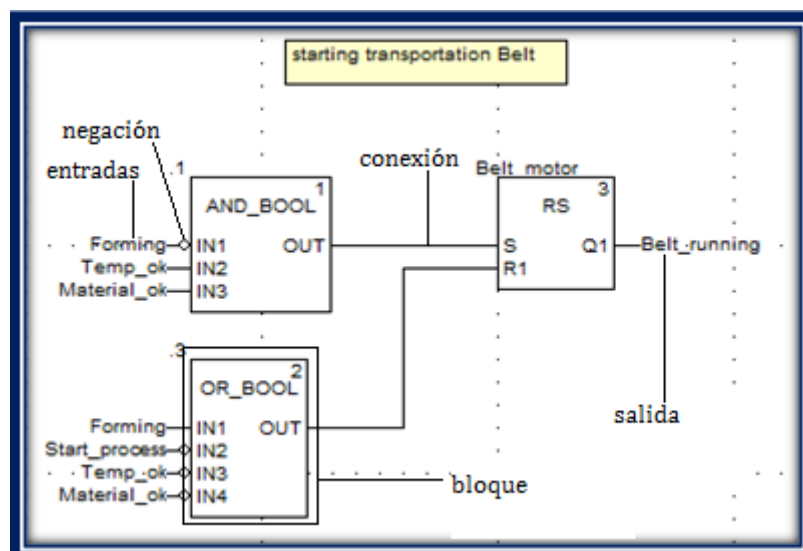


Figura 6: Ejemplo de programación FBD

Centraremos más la atención en los dos últimos lenguajes por ser los utilizados en la realización del proyecto. La estructura del diagrama de contactos es similar a un circuito de corriente para conmutadores de relé, usa bobinas, contactos y todos los bloques FFB anteriormente nombrados, así como elementos de control y unión. Dentro de las bobinas y contactos tenemos una amplia gama con diferentes funcionamientos como, por ejemplo, con retardos de activación, de flanco de subida/bajada, contactos y bobinas de llamada, bobinas cerradas/abiertas, etc... . Como en lenguajes anteriores, la secuencia del programa se ejecuta de arriba a abajo y el flujo de corriente va de izquierda a derecha (contactos) a derecha (bobinas). En la figura 7, vemos a modo de ejemplo un extracto del sistema de subida/bajada de las persianas del proyecto.

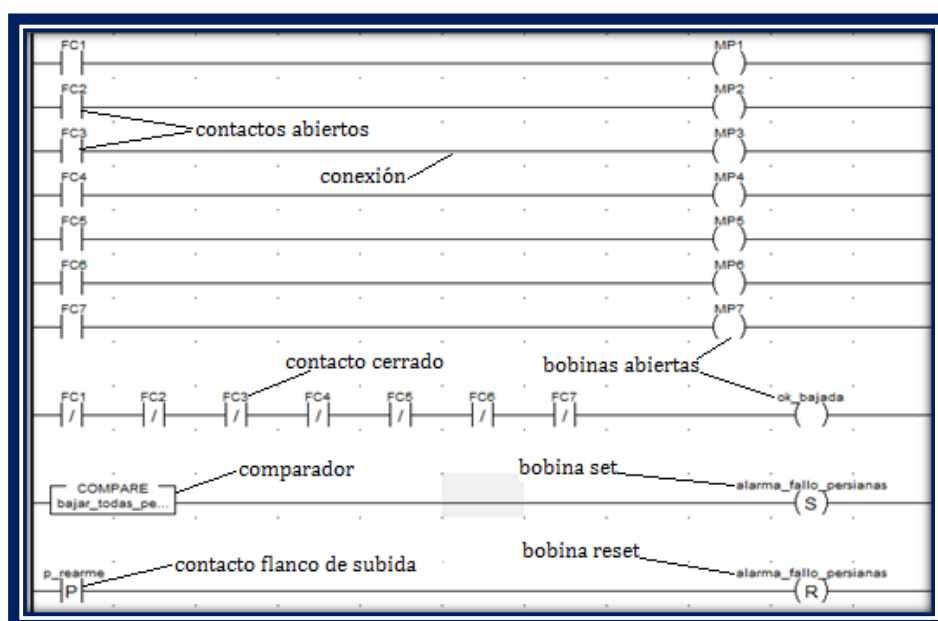


Figura 7: Ejemplo de programación LD

Por último, queda hablar del diagrama secuencial o SFC, este tipo de programación es la utilizada mayoritariamente en la realización del proyecto, sin embargo, la programación interna de las etapas y transiciones del SFC fue realizada en lenguaje de contactos. El SFC se comporta como un diagrama de flujo en el que se pueden organizar subrutinas, éstas pueden ser programadas en uno o varios de los otros 4 tipos de lenguajes disponibles, es sencillo y muy útil en aplicaciones que funcionan de manera secuencial. Un programa en SFC tiene dos elementos básicos, pasos (etapas) y transiciones (condiciones para entrar a etapa o salir de ella), pero también puede usar otro tipo de elementos como, por ejemplo, paso a macro (subcadena de pasos), bifurcación y conjunción alternativa (camino alternativo), bifurcación y conjunción simultánea (varios caminos realizados de manera simultánea) y conexiones y saltos.



En la figura 8, vemos el SFC general del proyecto en el que podemos observar muchos de los elementos que se pueden utilizar en la realización del SFC.

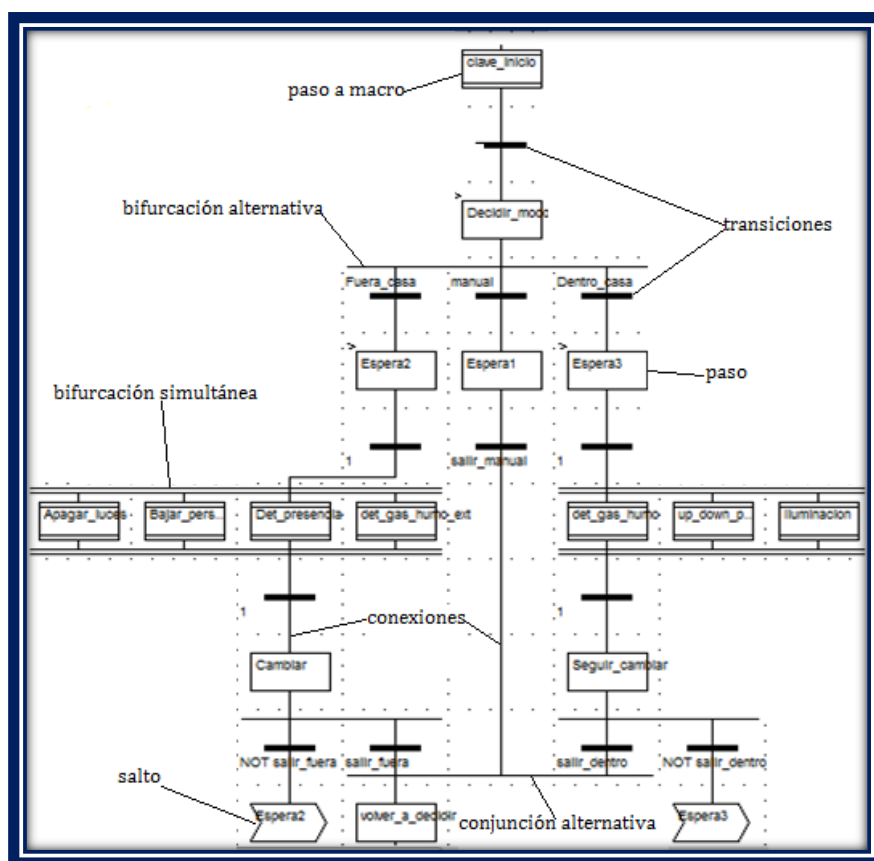


Figura 8: Ejemplo de programación SFC

El entorno de programación que incorpora Unity Pro es un entorno sencillo, intuitivo y de fácil aprendizaje que ayuda a la creación de programas de manera fluida.

Para empezar a programar dentro de nuestro proyecto, vamos al “explorador de proyectos” y desplegando la pestaña “programa” observamos a primera vista los submenús escalonados, “Tareas”-> “Mast”-> “Secciones”, nos centraremos en la tarea “Mast” (Máster, creada por defecto con nuestro proyecto), donde realizaremos nuestro programa, si fuera necesario se podrían crear más tareas diferentes a la “Mast” con el objetivo de tener una mejor organización. Para crear una sección, pinchamos con el botón derecho en Secciones-> Nueva Sección, a continuación se abrirá una pantalla como la que vemos en la figura 9, en la que podremos elegir entre otras opciones el nombre de dicha sección y el tipo de lenguaje que queremos usar para programar.

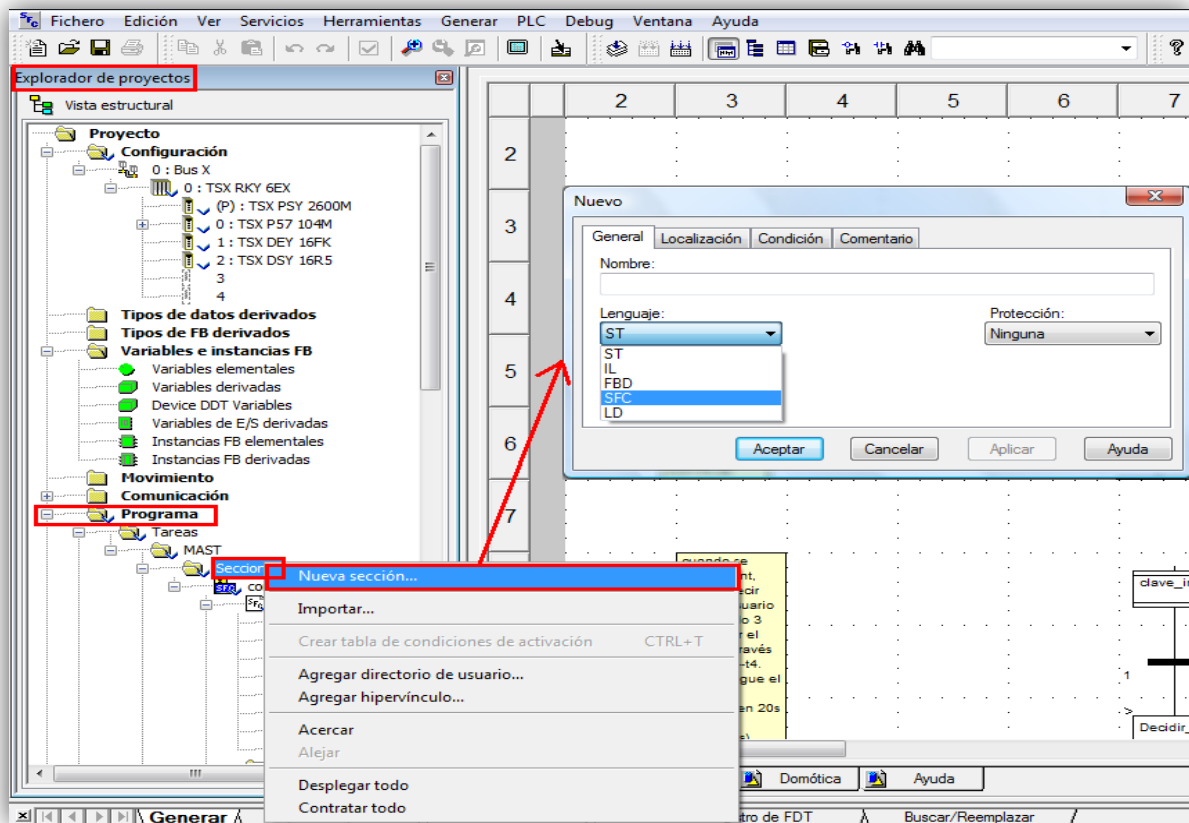


Figura 9 : Creación de una sección de programación

Cuando tenemos creada nuestra sección, nos aparecerá un entorno como el mostrado en la figura 10, en la parte superior de la imagen tenemos diferentes tipos de barras de herramientas, en el caso particular del ejemplo, la sección se ha creado en lenguaje SFC y aparece su barra de edición, si hubiera sido creada en cualquier otro, aparecería la correspondiente. Dentro de las barras de herramientas que aparecen, hay que destacar tres de ellas, la primera es la comentada anteriormente, el editor- SFC (o IL, LD, FBD, ST en cada caso), con el cual, podemos incluir todo tipo de conexiones, bifurcaciones y elementos específicos de cada lenguaje; después, la barra de servicios, donde encontramos botones para analizar el proyecto en busca de fallos, regenerar el proyecto, abrir la librería de tipos, etc..., y por último, la barra Api, que sirve para transferir los proyectos desde pc-autómata y viceversa, conectar/desconectar el autómatas y ponerlo en run/stop.

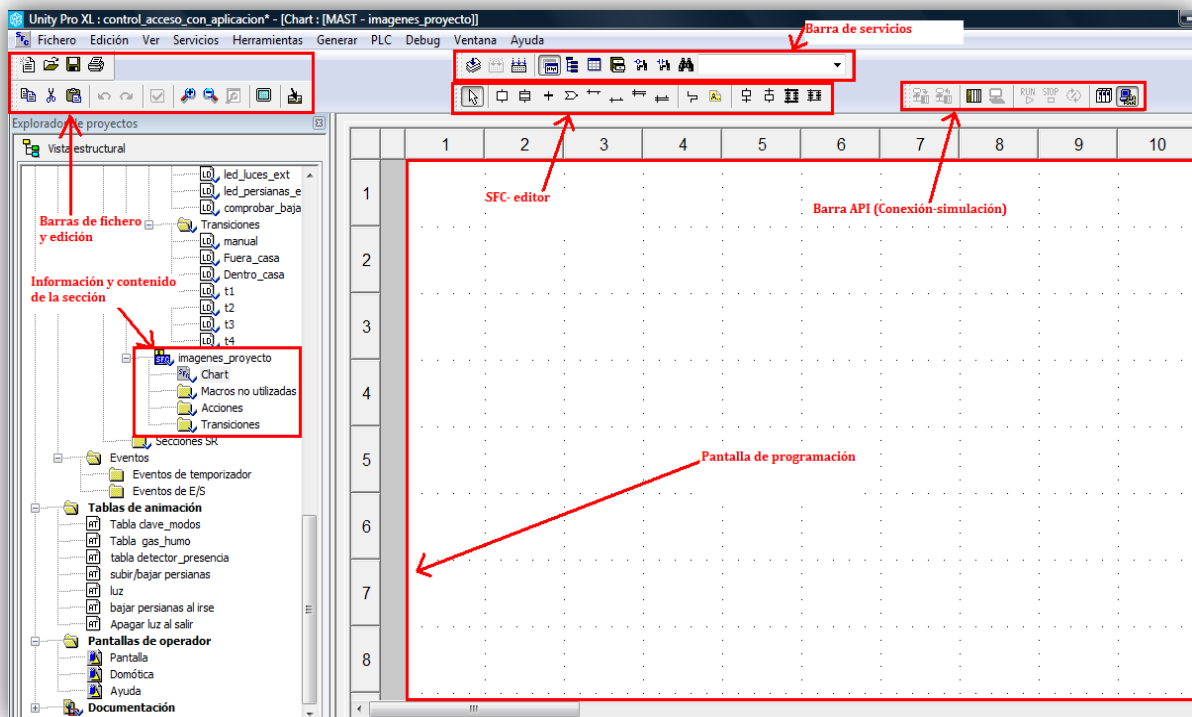


Figura 10: Entorno de programación Unity Pro XI

### 3.3. Tablas y entorno de simulación

Las simulaciones sin autómatas se pueden realizar de dos maneras, por medio de tablas de variables o por pantallas de operador, cada una tiene sus ventajas e inconvenientes pero las dos desempeñan la misma función.

Para crear una tabla de variables, dentro del “explorador de proyectos” pinchamos con el botón derecho en “Tablas de animación”-> “Nueva Tabla de animación”, en la ventana que se abre automáticamente, elegimos el nombre. Dentro de la pantalla creada, si pinchamos en el icono de puntos suspensivos que vemos en la figura 11, podremos incluir las variables que deseamos controlar, además, las tablas tienen dos modos de funcionamiento, modificación y forzado, y los iconos que vemos en la parte superior sirven para forzar/modificar valores de las variables durante el proceso de simulación.

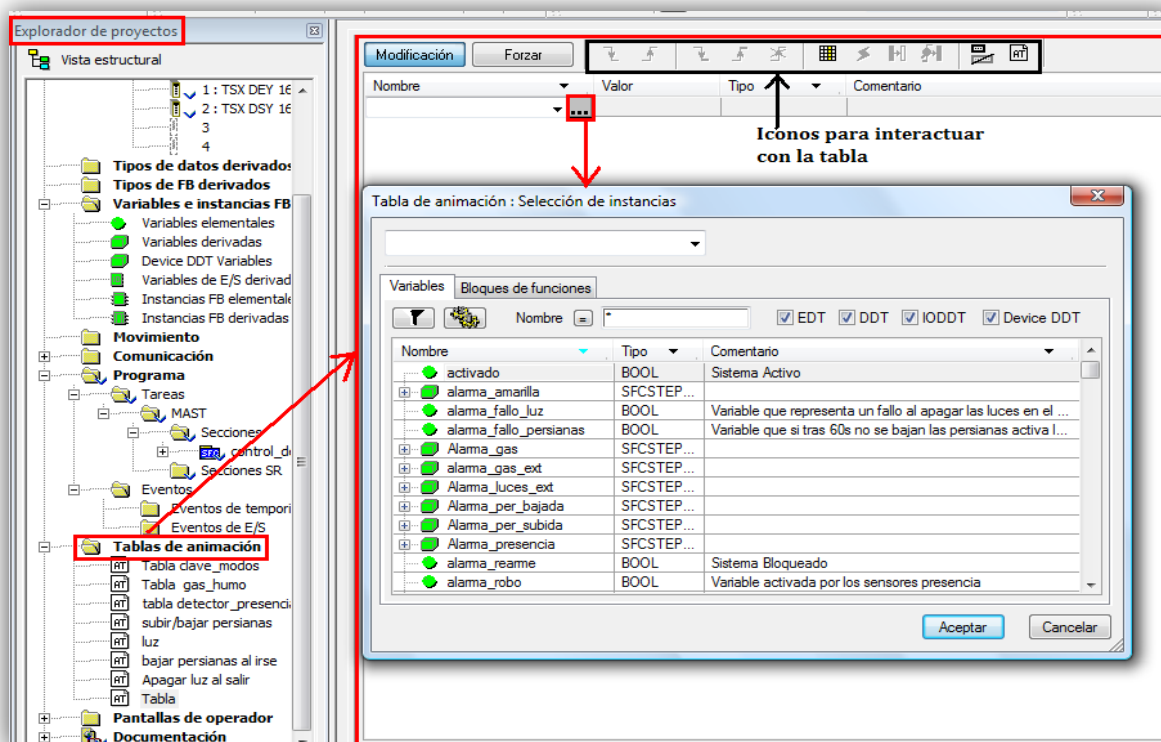


Figura 11: Creación de tabla de animación

De igual forma, pinchando en “Pantallas de operador” se crea una nueva pantalla que contiene la barra de herramientas IOSEditor, con ella se pueden introducir botones de comando, leds de alarma, casillas de verificación, imágenes, etc..., en la figura 12, podemos ver un ejemplo del entorno, sin embargo, en el capítulo 5 se profundizará sobre la creación y el diseño de la pantalla de operador y de la tabla de animación, ambas utilizadas en el proyecto.

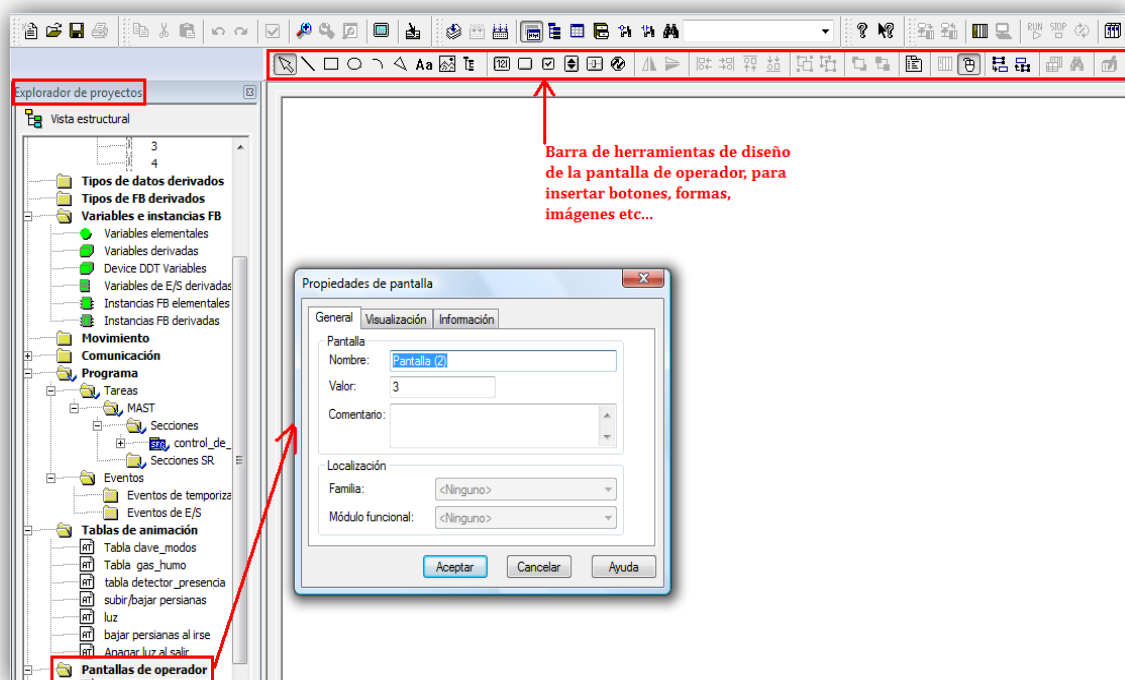


Figura 12: Creación de pantalla de operador

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 4: DISEÑO Y SOLUCIÓN DEL PROBLEMA  
PROPUESTO***



## 4.1. Presentación del problema

Como se ha visto en el Capítulo 2, la domótica tiene fundamentalmente tres áreas funcionales dentro de la vivienda, confort, comunicaciones y seguridad, en nuestro caso nos centraremos en confort y seguridad. La razón radica, en que ambas son las áreas más importantes en un sistema domótico, también, el hecho de disponer únicamente de un módulo de 16 entradas y salidas en el autómata dificulta la elección y limita el número de sistemas a incluir, esto se debe principalmente, a que los sistemas utilizados en domótica están distribuidos a lo largo de la vivienda y necesitan múltiples sensores y actuadores (con la necesidad de entradas y salidas al autómata que ello conlleva).

La implantación de nuestro sistema se centrará en realizar el control automático de las persianas e iluminación (confort) y la seguridad ante fugas de gas, humo e intrusión en el interior de la vivienda. El objetivo será controlar de manera automática y eficiente los sistemas para mejorar el confort y seguridad de los residentes de la vivienda, mejorando a su vez, el rendimiento energético de la misma y aprovechando al máximo las horas de luz natural. Además, buscamos diseñar un sistema que cumpla varias condiciones imprescindibles:

- Minimizar el coste, reutilizando sensores y actuadores para diferentes usos o sistemas.
- El simulador debe ser sencillo e intuitivo y mostrar en todo momento el estado de persianas, luces y alarmas, de manera que sea fácilmente entendible para el usuario.
- La programación de la solución debe ser fácilmente ampliable para poder incorporar posteriormente más sistemas (calefacción, comunicaciones...) sin necesidad de rediseñar toda la aplicación.
- Debe informar de errores en el funcionamiento del dispositivo.

## 4.2. Diseño de la solución

### 4.2.1. Estructura general, funcionamiento y planificación del diseño

En el diagrama de flujo de la figura 13, se observa la estructura general del diseño planteado, en apartados posteriores se explicará detalladamente cada uno de los subsistemas (macroetapas) que lo forman.

El sistema comienza con una clave de seguridad de 4 dígitos que debe ser introducida para poder activarlo, si el usuario introduce mal un número o tarda más de 20 segundos en pulsar, se considerará como error, al tercer error cometido, se producirá la activación de una alarma con su consecuente bloqueo, para desbloquearlo, es necesario pulsar un botón de rearme (en la realidad, el botón será sustituido por una llave para mayor seguridad).

Una vez activado, se han establecido 3 modos de funcionamiento con el objetivo de aportar mayor flexibilidad en su uso ante diferentes situaciones, por tanto, se puede elegir entre:

- **Manual:** Todos los sistemas se encuentran inactivos y el autómatas no utiliza ni sensores y actuadores, la vivienda se comporta de manera convencional.
- **Fuera de casa:** Ha sido creado para apagar determinados sistemas (bajar persianas y apagar luces) y activar otros (detector gas/humo e intrusión) de manera simultánea en salidas rápidas de casa.
- **Dentro de casa:** Funcionan automáticamente y de manera simultánea todos los sistemas implantados excepto la detección de intrusión, además, tiene la posibilidad de apagar cualquier sistema que no se quiera utilizar, por ejemplo, se quiere que funcionen las alarmas de gas y humo, pero no el control automático de luces y persianas.

Dentro ya de cualquier modo, entramos en un ciclo de ejecución, cuando termina un ciclo se comprueba que el usuario no tiene intención de cambiar de modo, esto se lleva a cabo comprobando la posición de los selectores de modo (se explicará en detalle en el capítulo 4), si ésta ha variado, se regresa de nuevo al estado de selección de modo, si no, se comienza un nuevo ciclo.

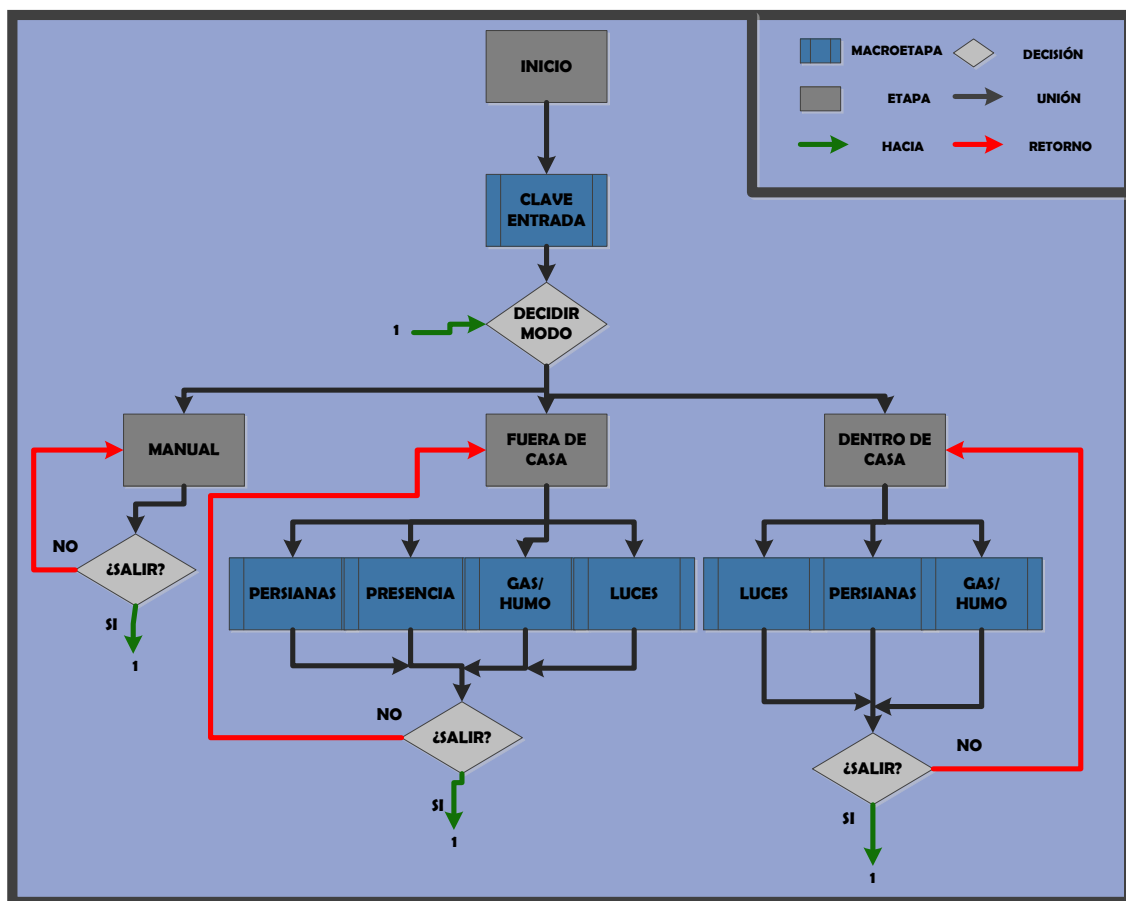


Figura 13: Diagrama de flujo del sistema principal

Con el funcionamiento general entendido y antes de pasar a explicar los subsistemas en detalle (representados con macroetapas), tenemos que hacer una previsión de la cantidad de sensores y actuadores que vamos a necesitar. El sistema se va a implantar únicamente en la planta baja de la vivienda, esta consta de 6 estancias y 7 ventanas en total (recibidor, salón, estudio, baño, cocina y garaje), en la siguiente lista se citan los dispositivos necesarios:

#### **Control de acceso (clave seguridad para activar sistema)**

- 4 pulsadores (del número 1-4) para poder introducir una clave de seguridad de 4 dígitos.
- 1 pulsador de rearme, el cual, se utilizará para rearmar todos los sistemas de alarmas y errores.
- 2 selectores para decidir el modo de funcionamiento entre manual/fuera de casa/dentro de casa.
- 1 alarma de fallo que se activa al cometer 3 fallos en la introducción de la clave de seguridad.
- 4 leds, uno para verificar el estado activo del sistema y 3 para el modo de funcionamiento.

#### **Control de iluminación**

- 1 sensor de luz exterior para controlar la cantidad de luz natural externa.
- 1 interruptor para encender/apagar el sistema.
- 6 sensores de presencia, uno para cada habitáculo.
- 6 interruptores (actuadores) para encender la luz en cada habitáculo.
- 1 alarma de fallo en el sistema de control de luz.

#### **Control de intrusión**

- 1 interruptor para encender/apagar el sistema.
- 6 sensores de presencia, uno para cada habitáculo.
- 1 alarma de intrusión o presencia en el interior de la casa.

#### **Control persianas**

- 1 interruptor para encender/apagar el sistema.
- 7 sensores final de carrera (uno por cada persiana).
- 7 motores de persiana (uno por cada persiana).
- 1 alarma de fallo en el sistema de control de persianas.
- 1 sensor de luz exterior para controlar la cantidad de luz natural externa.



### **Control gas/humo**

- 1 interruptor para encender/apagar el sistema.
- 1 sensor de gas.
- 1 sensor de humo.
- 1 alarma gas o humo.
- 1 electroválvula de gas.

Contando con la incorporación de todos estos sistemas, nuestro programa necesitará como mínimo el siguiente número de entradas y salidas disponibles en el autómata para poder funcionar correctamente:

Entradas	Salidas
27	23

**Tabla 1: Número mínimo de entradas y salidas necesarias**

## **4.2.2. Diseño y funcionamiento de los subsistemas**

En este apartado, se entrará en detalle en el funcionamiento de cada uno de los subsistemas comentados anteriormente en la estructura general del sistema, se llevará a cabo con ayuda de diagramas de flujo para facilitar el entendimiento.

### **4.2.2.1. Clave de seguridad**

Este subsistema ha sido diseñado para aportar seguridad y un grado de realismo al sistema. Su funcionamiento consiste en introducir una clave predeterminada de 4 dígitos, hasta que el usuario no haya tecleado correctamente el primer dígito no se puede introducir el segundo y así sucesivamente (sin referencia visual del número que está introduciendo en cada momento, con el objetivo de no ayudar a posibles robos); si se equivoca al teclear un número o tarda demasiado tiempo (más de 20 segundos), se contará como error y se volverá al inicio; al producirse 3 errores el sistema se bloqueará y saltará una alarma para avisar de una incidencia. El desbloqueo se realiza mediante un pulsador de rearme, el cual, en una aplicación real sería una llave de alta seguridad que sólo el dueño de la vivienda dispondría.

En el diagrama de flujo representado en la figura 14, se puede apreciar el funcionamiento del sistema, también, podemos observar que una vez activado el sistema se pasa directamente a poder elegir el modo de funcionamiento.

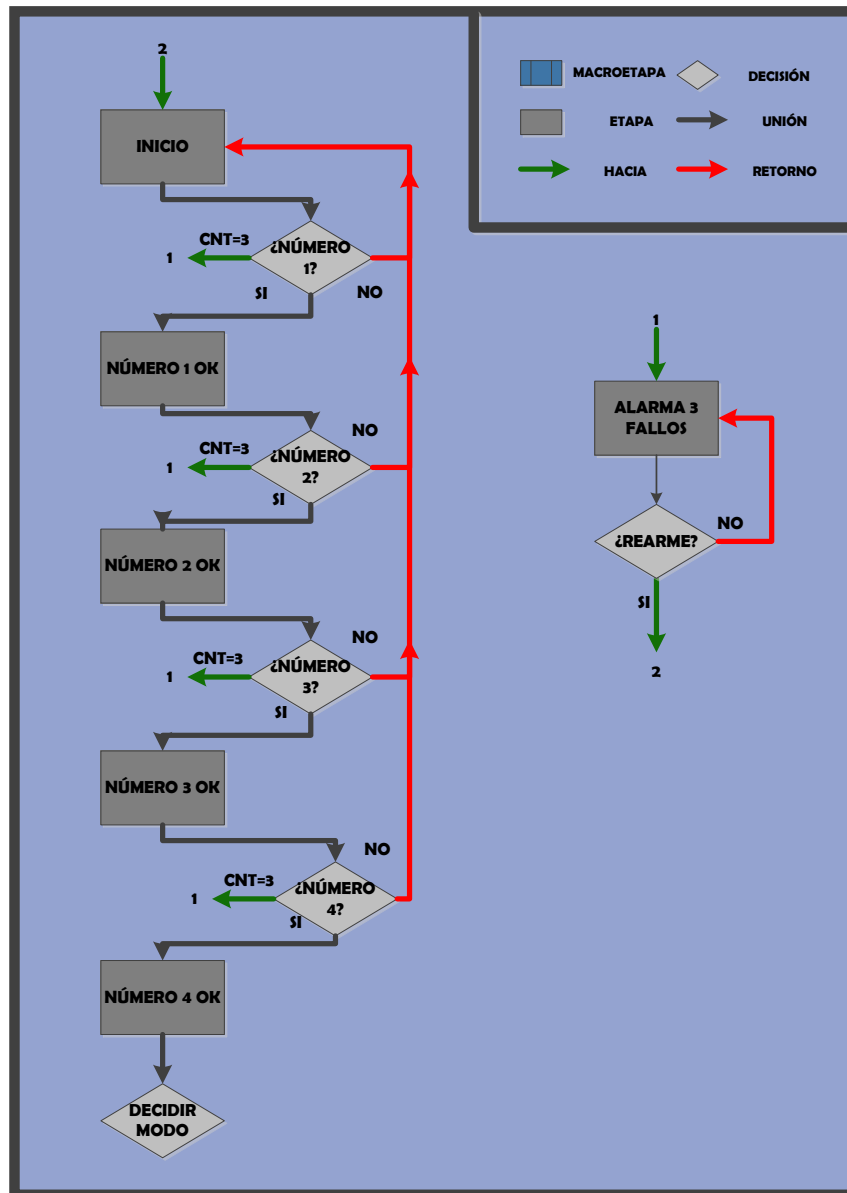


Figura 14: Diagrama de flujo de sistema clave de seguridad

#### 4.2.2.2. Iluminación

El sistema de control automático de luces funciona en los modos: fuera y dentro de casa, en cada uno tiene un funcionamiento diferente y han sido diseñados y programados de manera independiente entre sí, aun así, comparten los sensores y actuadores. Este sistema no entra en la cantidad de intensidad de luz que emiten los focos colocados en la casa, únicamente se centra en la conmutación automática con determinadas condiciones externas e internas. Se explicará cada uno por separado:

### Iluminación en modo fuera de casa

El objetivo de este sistema es apagar las luces sin tener en cuenta las condiciones ambientales y avisar en caso de que se produzca algún problema al realizar la ejecución.

En primer lugar, se comprueba que todas las luces estén apagadas, si no es así, se procede a apagarlas, además, se ha implementado un sistema de seguridad, el cual, detecta si no se han apagado las luces en un periodo de tiempo (1 minuto después) desde que se ejecutó la orden, si esto ocurre, se activa una alarma luminosa/sonora que avisa al usuario de la existencia de una incidencia.

Este sistema se rearma de la misma manera que la clave de seguridad, una vez pulsado el botón de rearme, se vuelve a realizar el ciclo desde el principio como se puede observar en la figura 15.

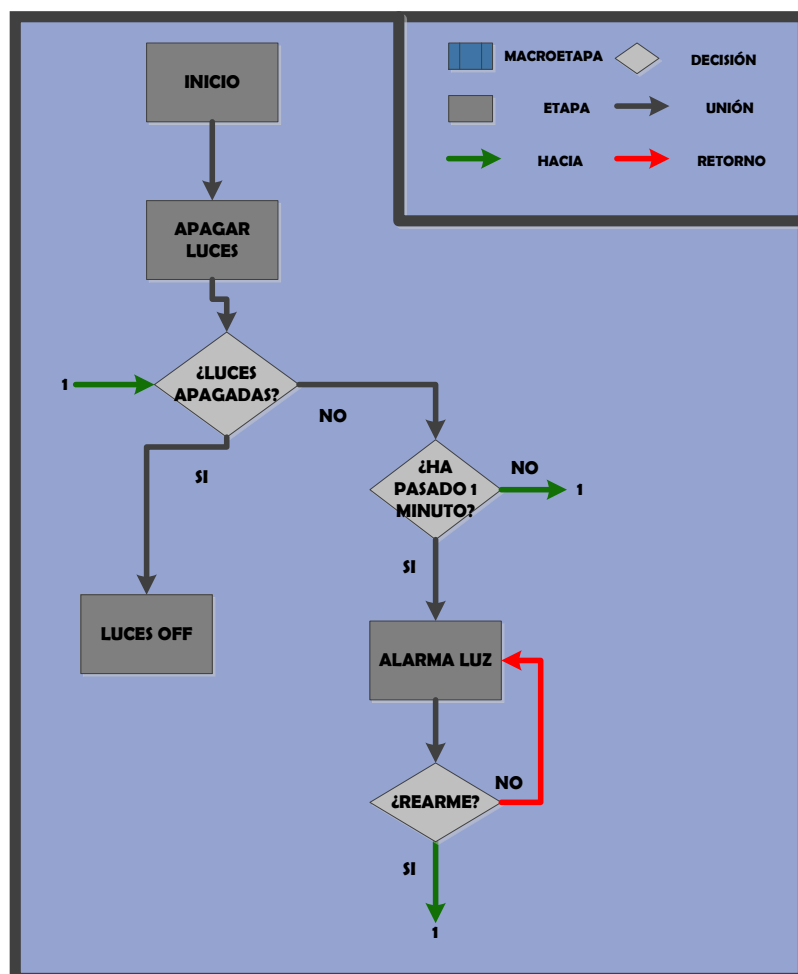


Figura 15: Diagrama de flujo de sistema de iluminación (fuera de casa)

### Iluminación en modo dentro de casa

El sistema para el modo dentro de casa funciona de manera diferente a lo explicado anteriormente, para empezar, este sistema tiene un interruptor que activa su funcionamiento, si no se encuentra activo no funcionará. En este caso, si se tienen en cuenta las condiciones ambientales, siendo medidas por un sensor de luz colocado en el exterior de la vivienda, el cual, detecta si la cantidad de luz es suficiente (si es así, solo se encenderán las luces del baño y garaje, que son los sitios con poca luminosidad) o hace falta encender las luces artificiales (toda la casa).

El encendido de las luces se realiza por medio de seis sensores de presencia, uno por habitáculo, se tiene en cuenta el estado del sensor de iluminación y de los sensores de presencia para encender las luces de las salas correspondientes. Éstas permanecen encendidas mientras el sensor detecte movimiento, si éste permanece más de 10 segundos sin captar presencia, la luz se apagará automáticamente hasta que vuelva a detectarla, con el fin de ahorrar energía cuando se está entrando y saliendo continuamente de las habitaciones. En la figura 16, se puede observar el funcionamiento del citado sistema.

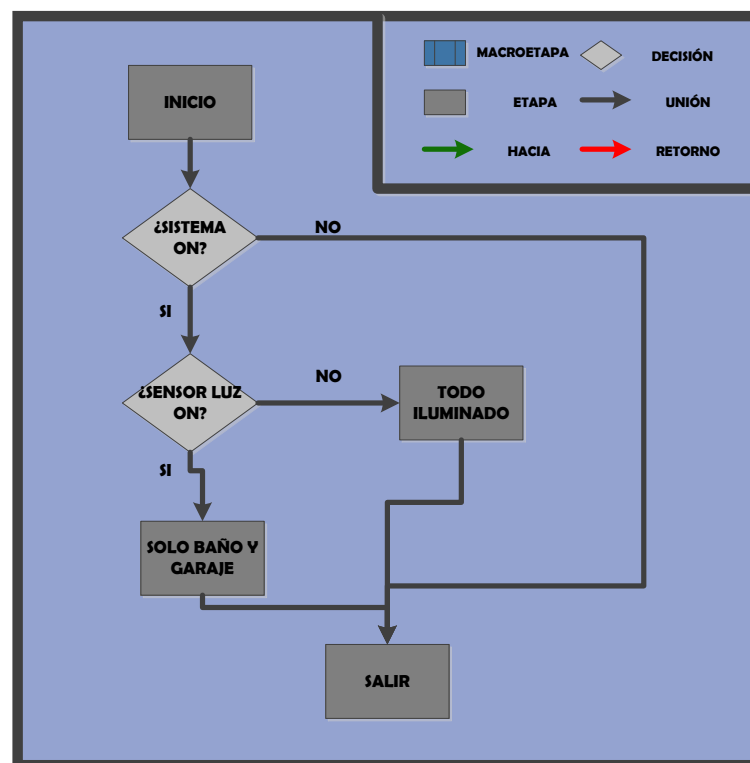


Figura 16: Diagrama de flujo de sistema de iluminación (dentro de casa)

#### **4.2.2.3. Persianas**

En el control automático de persianas sucede lo mismo que en la iluminación, se han realizado dos programaciones diferentes, una para el modo dentro de casa y otra para el modo fuera de casa, y de igual manera, se aborda cada una por separado. También, hay que destacar que sólo se han considerado dos posibilidades a la hora de establecer automáticamente la posición de las persianas, bajadas o subidas.

##### **Control de persianas en modo fuera de casa**

El objetivo de este sistema es análogo al explicado anteriormente para la iluminación, en este caso se busca que las persianas bajen automáticamente al entrar en dicho modo.

En primer lugar, se comprueba si todas las persianas están bajadas (esto se realiza con sensores final de carrera, los cuales, detectan la posición de cada una de las persianas), si no, se da la orden de bajada (con ella se activan los motores de cada persiana para que cambien de posición). De igual forma que en la iluminación, si ha pasado 1 minuto desde que se dio la orden y algún sensor final de carrera indica que la posición no es la adecuada, se lanzará una alarma de incidencia (luminosa/sonora) en el funcionamiento del sistema. Como en sistemas anteriores, el pulsado del botón de rearme hace que se vuelva a realizar el ciclo desde el principio.

Si miramos el diagrama de la figura 17, vemos que el punto de retorno que establece el pulsador de rearme se encuentra en el estado que comprueba si las persianas están bajadas, con lo que en realidad se está realizando un bucle del que sólo se saldrá si las persianas alcanzan la posición esperada.

##### **Control de persianas en modo dentro de casa**

El funcionamiento entre este modo y el anterior es parecido pero tiene diferencias, la primera de ellas es que este sistema se puede inhabilitar mediante un interruptor de apagado, esto se ha introducido para dar flexibilidad en su uso y poder disponer del funcionamiento únicamente cuando el usuario desee utilizarlo. Otra diferencia es el uso del sensor de iluminación exterior, el cual, será muy útil a la hora de decidir bajar o subir las persianas de la casa según las condiciones de luminosidad captadas por el sensor.

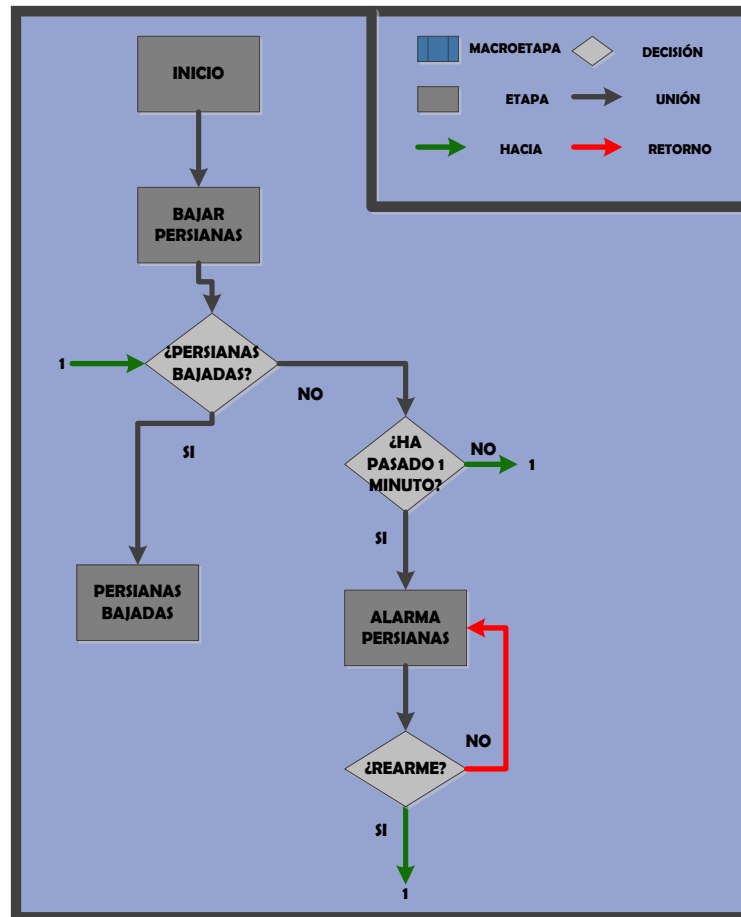


Figura 17: Diagrama de flujo de sistema de persianas (fuera de casa)

Si observamos el diagrama de flujo representado en la figura 18, vemos que primero se examina si el sistema está habilitado, si no lo estuviera, se saldría del mismo. Cuando el sistema está activo, se comprueba el estado del sensor de iluminación exterior, si este se encuentra a 1 quiere decir que hay luz suficiente y se procede a la subida de las persianas, o, en su defecto, a verificar si están todas subidas.

Si el sensor no detecta luminosidad suficiente (se encuentra a 0), se procede a comprobar si las persianas están bajadas, y si no fuera así, activaría los motores correspondientes para colocarlas en la posición correcta.

El sistema de alarma implementado se rige por las mismas reglas que para el modo fuera de casa, si en un periodo de tiempo de 1 minuto y habiendo dado la orden de bajada o subida de las persianas, se capta que alguna persiana no está en la posición esperada, saltará de nuevo la alarma de incidencia. Como hemos visto en todos los sistemas anteriores, para salir de este estado de alarma necesitamos pulsar el botón de rearme y se volverá a realizar otro ciclo de ejecución.

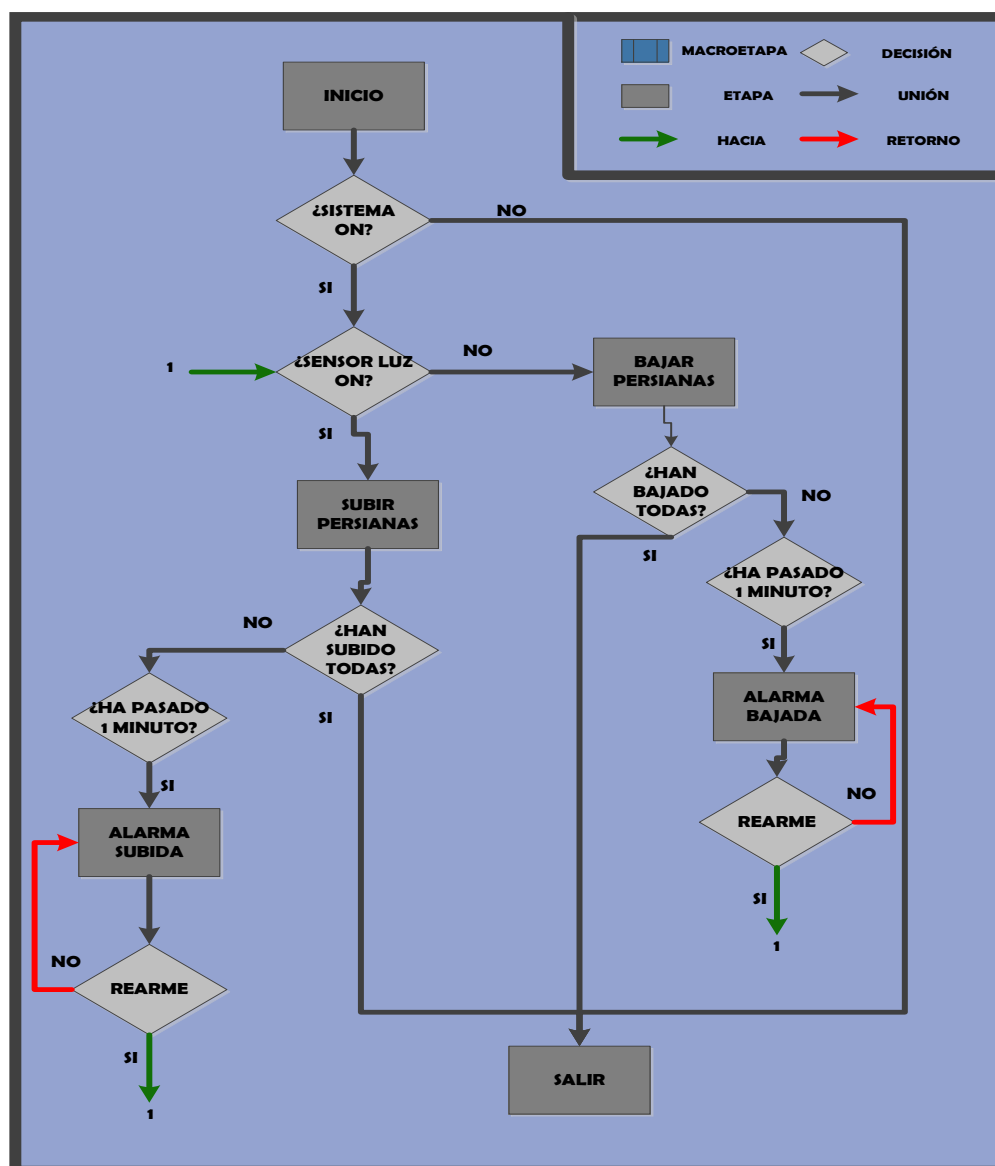


Figura 18: Diagrama de flujo de sistema de persianas (dentro de casa)

#### 4.2.2.4. Detección de gas y humo

En los dos subsistemas dedicados al área funcional de la seguridad, vamos a poder habilitar o no las alarmas dependiendo de la necesidad del usuario mediante un botón de encendido.

El sistema de detección de gas y humo funciona igual para los modos dentro y fuera de casa, se trata de una medida de seguridad adicional para asegurar una rápida actuación ante situaciones peligrosas, tales como una fuga de gas o un posible incendio, y que el usuario pueda reaccionar a tiempo.

Lo primero que se comprueba es si el sistema está habilitado (si no se sale automáticamente del sistema), si es así, se observa el estado de los sensores de gas y humo, si alguno de éstos se activa (cualquiera de los dos o los dos, pasan de 0 a 1), automáticamente se cerrará la electroválvula de gas (para evitar una posible explosión) y se encenderá una alarma para alertar a los inquilinos de la vivienda de la existente situación. Se volverá a un estado normal de funcionamiento cuando se haya pulsado el botón de rearme, el cual, sirve para salir del estado de alarma (se supone que el usuario lo ha pulsado porque ha subsanado el error detectado por el sistema y quiere que vuelva a funcionar con normalidad).

En la figura 19, se puede apreciar gráficamente cómo funciona la secuencia de ejecución del sistema de alarma de gas y humo.

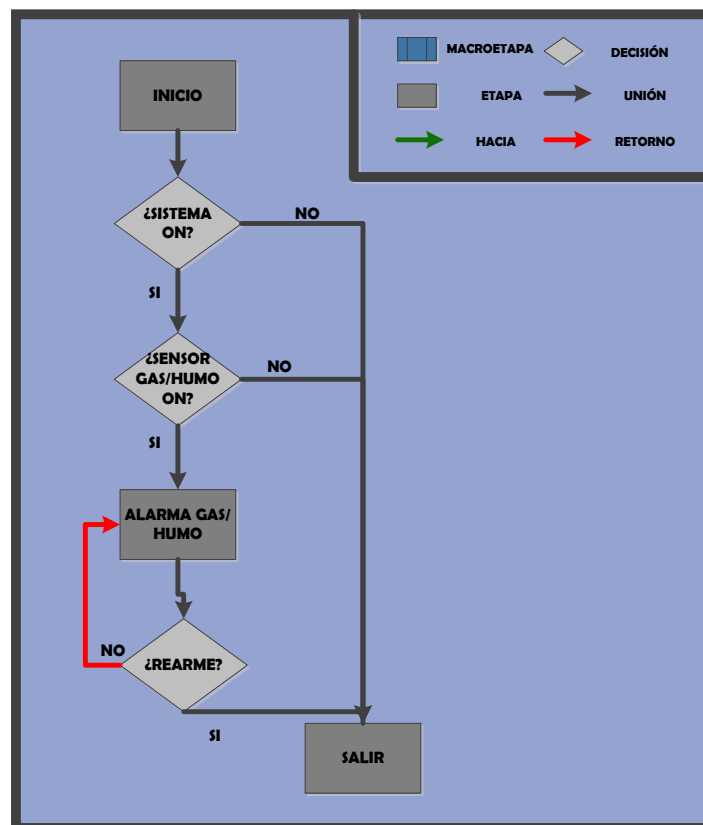


Figura 19: Diagrama de flujo de sistema de detección de gas y humo



#### 4.2.2.5. Detección de presencia

El último sistema que queda por explicar es la detección de presencia en el interior de la casa, ésta sólo estará disponible en el modo fuera de casa. Consiste en una alarma que utiliza los sensores de presencia para detectar movimiento en el interior de la casa cuando no se encuentra nadie en la vivienda. Este sistema es susceptible a fallos si en la vivienda habitan animales de compañía y el usuario activa la alarma dejando a su mascota dentro de casa, ya que, como se puede imaginar, los detectores de movimiento captarán la situación y activarán la alarma, por ello, se ha dotado al sistema de la posibilidad de habilitarlo o no, dependiendo de las necesidades de la persona.

Observando el diagrama contenido en la figura 20, lo primero que se comprueba es si el sistema está habilitado, si es así, se examinan constantemente los sensores de presencia, y si en algún instante cualquiera de ellos detecta movimiento, se activará automáticamente una alarma de intrusión, con el objetivo de alertar al propietario y de disuadir al posible ladrón.

Existen muchas posibles mejoras en este sistema de seguridad, el cual, se ha implementado de manera más sencilla debido a la limitación de entradas y salidas del autómat. En este subsistema se podría haber incorporado un módulo de comunicaciones/seguridad que tuviera un control del perímetro de la casa, incorporando seguridad de ventanas y puertas exteriores (sensores efecto hall), intrusión en zonas ajardinadas exteriores (sensores de presencia) y comunicación a la policía y al usuario, vía internet, sms, teléfono, etc...

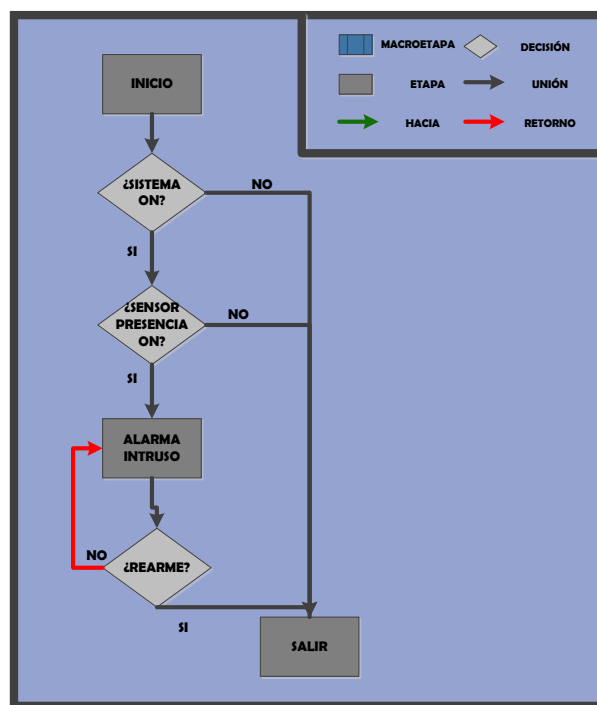


Figura 20: Diagrama de flujo de sistema de detección de presencia

### 4.3. Programación en Unity Pro XI

#### 4.3.1. Configuración de Unity Pro XI

El autómatas utilizado para la realización de este trabajo es el disponible en el laboratorio de automática (ver figura 21), el Premiun TSX de Scheneider, está compuesto de los siguientes elementos:

- Bastidor
- Fuente de alimentación
- Procesador
- Módulo de entradas digitales
- Módulo de salidas digitales
- Módulo de entradas analógicas
- Módulo de salidas analógicas
- Telefast
- Cable comunicación con PC



Figura 21: Autómatas Premiun TSX del laboratorio

Para comenzar a trabajar hay que configurar Unity Pro correctamente, esto consiste en incluir el modelo de autómatas y los modelos concretos de los diferentes módulos de hardware instalados en el autómatas, entre ellos, se encuentran la fuente de alimentación y los módulos de entradas y salidas digitales. En Unity Pro, esto se realiza de dos maneras:

1. En la creación del proyecto, elegimos el modelo de autómatas con el que vamos a trabajar. Si observamos la figura 22, vemos que al crear un nuevo proyecto se abre una ventana en la que aparece la posibilidad de elegir entre las familias de autómatas Modicon, Premium y Quantum, a su vez, dentro de éstas podremos elegir modelos específicos de cada una, en nuestro caso, elegiremos la familia Premium y el modelo concreto “TSX P57 104M 0260 57-1”.

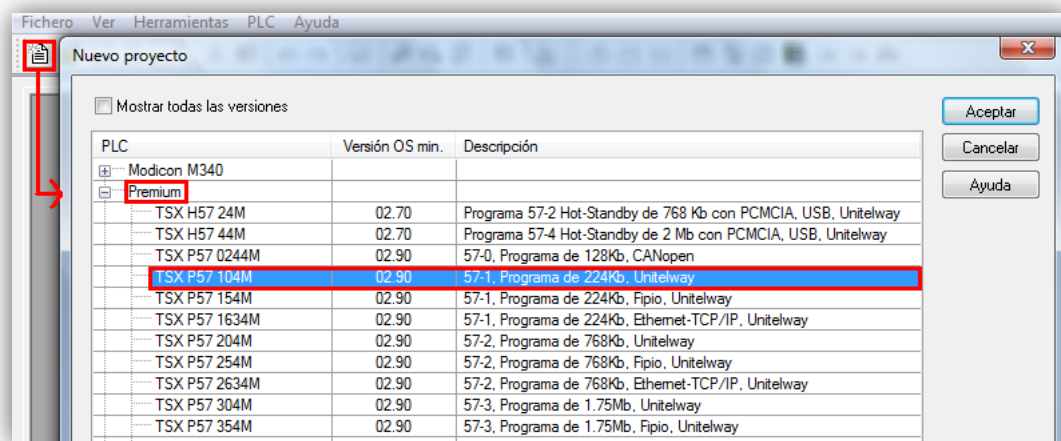


Figura 22: Creación nuevo proyecto en Unity Pro XI

2. Configuramos el hardware necesario para trabajar con el autómata, dentro de nuestro proyecto nos dirigimos a explorador de proyectos-> configuración, aquí observamos que por defecto se coloca un único bastidor de 6 posiciones, el "TSX RKY 6EX", si pinchamos sobre él se abrirá una imagen como la mostrada en la figura 23, donde aparecen configurados por defecto los dos primeros slots (posiciones) del rack, veamos en detalle lo que se puede incluir en cada posición y lo que se ha incluido en nuestro caso:

1. Posición 0: Se coloca por defecto la fuente de alimentación modelo: TSX PSY 2600M, si no apareciera, se pincharía en el slot y se introduciría dicho modelo.
2. Posición 1: Es la posición de la CPU correspondiente a cada autómata, se introduce por defecto al crear el proyecto, contamos con el modelo: TSX P57 104M 57-1.
3. Posición 2: En las demás posiciones, se da libertad para poder incluir todo tipo de módulos, en la figura 23, vemos que al pinchar sobre cualquiera de estas posiciones se despliega una pantalla con diferentes opciones para poder elegir. En este caso, nosotros elegiremos un bloque de 16 entradas digitales: DEY 16 FK.
4. Posición 3: Aquí seleccionamos un módulo complementario al anterior de 16 salidas digitales: DSY 16R5.

Las posiciones 4 y 5 quedan libres como se ve en la figura 23, ya que no necesitaremos de su uso para este proyecto.

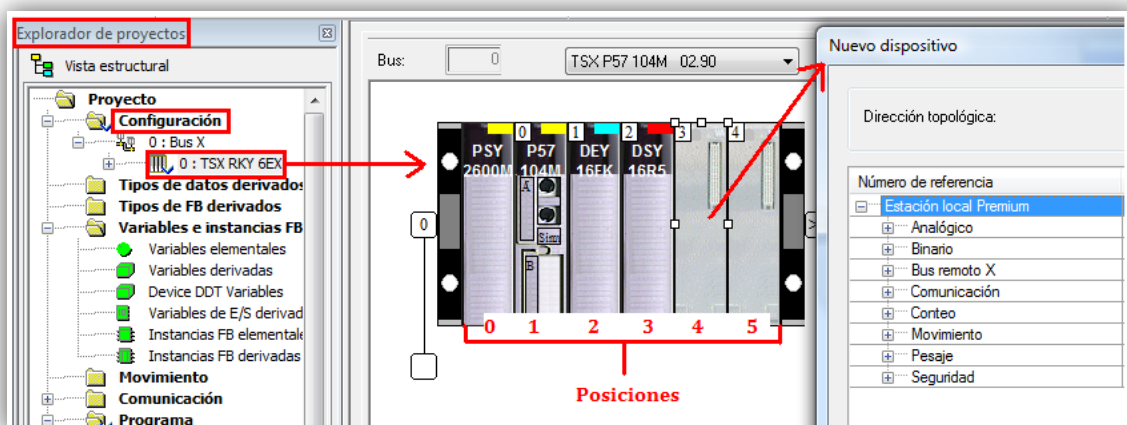


Figura 23: Configuración hardware del autómata

### 4.3.2. Definición de las variables necesarias

La definición de las variables es de suma importancia para nuestro programa, debido a que se usarán tanto internamente durante la ejecución del programa, como para modificar entradas y salidas del autómata, siempre con el objetivo de conseguir controlar el sistema.

Dentro del Explorador de proyectos nos vamos a Variables e instancias FB, aquí aparecerán varios tipos (ver figura 23, arriba) de los que utilizaremos:

- Variables elementales: Variables que declararemos para usar en simulación, y, también, como variables de entrada/salida.
- Variables derivadas: Cada etapa creada tiene una variable como esta asociada, la cual, contiene en su interior 4 datos (t, x, tminErr y tmaxErr) que pueden ser usados para controlar diferentes parámetros internos de la etapa como, por ejemplo, el tiempo.
- Instancias FB elementales: Aquí se encuentran emplazados los temporizadores y contadores que usaremos en el programa.

Para introducir una nueva variable o modificar una existente, simplemente pinchamos en “Variables elementales” y se abrirá una pantalla como la que vemos en la figura 24, en la que tendremos que introducir el nombre de la variable, el tipo (Bool, EBool, int, etc...), la dirección en el autómata y un comentario (opcional).

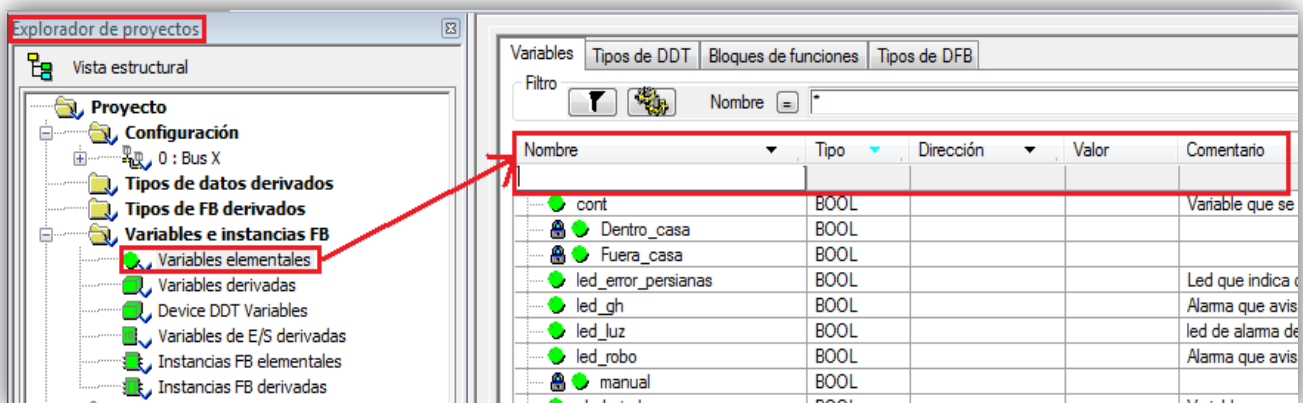


Figura 24: Creación de variables

En la figura 25, se ven reflejadas todas las variables elementales usadas en nuestro programa, vemos que únicamente son usados los tipos BOOL y EBOOL, este último, es usado para las variables que intervienen como entradas y salidas en la simulación, mientras que las BOOL son usadas de manera interna en el programa.

• alarma_fallo_luz	BOOL	• FC2	EBOOL	• p2	EBOOL
• alarma_fallo_persianas	BOOL	• FC3	EBOOL	• p3	EBOOL
• alarma_robo	BOOL	• FC4	EBOOL	• p4	EBOOL
• aviso_gh	BOOL	• FC5	EBOOL	• p_reame	EBOOL
• cont	BOOL	• FC6	EBOOL	• per_on	EBOOL
• Dentro_casa	BOOL	• FC7	EBOOL	• sensor_gas	EBOOL
• Fuera_casa	BOOL	• fuera_led	EBOOL	• sensor_humo	EBOOL
• manual	BOOL	• IL1	EBOOL	• sensor_luz	EBOOL
• ok_bajada	BOOL	• IL2	EBOOL	• SP1	EBOOL
• ok_luces	BOOL	• IL3	EBOOL	• SP2	EBOOL
• ok_subida	BOOL	• IL4	EBOOL	• SP3	EBOOL
• salir_dentro	BOOL	• IL5	EBOOL	• SP4	EBOOL
• salir_fuera	BOOL	• IL6	EBOOL	• SP5	EBOOL
• salir_gh	BOOL	• in_out	EBOOL		
• salir_luces_ext	BOOL	• led_error_persianas	EBOOL		
• salir_manual	BOOL	• led_gh	EBOOL		
• salir_persianas_ext	BOOL	• led_luz	EBOOL		
• salir_presencia	BOOL	• led_robo	EBOOL		
• t1	BOOL	• luz_on	EBOOL		
• t2	BOOL	• manual_led	EBOOL		
• t3	BOOL	• MP1	EBOOL		
• t4	BOOL	• MP2	EBOOL		
• activado	EBOOL	• MP3	EBOOL		
• alarma_reame	EBOOL	• MP4	EBOOL		
• dentro_led	EBOOL	• MP5	EBOOL		
• det_gh_on	EBOOL	• MP6	EBOOL		
• det_pre_on	EBOOL	• MP7	EBOOL		
• EV_gas	EBOOL	• on_off	EBOOL		
• FC1	EBOOL	• p1	EBOOL		

Figura 25: Variables elementales utilizadas en el sistema

Podemos observar que algunas variables cuentan con un candado a su izquierda, esto se debe, a que se crean automáticamente al introducir el nombre de una transición dentro de una determinada sección de programación.

Además, tenemos que diferenciar entre variables de entrada y salida, en las figuras 26 y 27, observamos de manera independiente las variables de entrada y salida que utilizaremos para la simulación de nuestro sistema, en el siguiente apartado se explicará con detalle qué función desempeña cada una y cómo funcionan dentro del programa.

• on_off	EBOOL	%I0.1.6	• det_gh_on	EBOOL	
• p1	EBOOL	%I0.1.0	• det_pre on	EBOOL	%I0.1.14
• p2	EBOOL	%I0.1.1	• FC1	EBOOL	
• p3	EBOOL	%I0.1.2	• FC2	EBOOL	
• p4	EBOOL	%I0.1.3	• FC3	EBOOL	
• p_reame	EBOOL	%I0.1.4	• FC4	EBOOL	
• per_on	EBOOL		• FC5	EBOOL	
• sensor_gas	EBOOL		• FC6	EBOOL	
• sensor_humo	EBOOL		• FC7	EBOOL	
• sensor_luz	EBOOL	%I0.1.15	• in_out	EBOOL	%I0.1.5
• SP1	EBOOL	%I0.1.7	• luz_on	EBOOL	%I0.1.13
• SP2	EBOOL	%I0.1.8	• SP5	EBOOL	%I0.1.11
• SP3	EBOOL	%I0.1.9	• SP6	EBOOL	%I0.1.12
• SP4	EBOOL	%I0.1.10			

Figura 26: Variables de entrada del sistema

● activado	EBOOL	● led_error_persianas	EBOOL
● alarma_rearme	EBOOL	● led_gh	EBOOL
● dentro_led	EBOOL	● led_luz	EBOOL
● EV_gas	EBOOL	● led_robo	EBOOL
● fuera_led	EBOOL	● manual_led	EBOOL
● IL1	EBOOL	● MP1	EBOOL
● IL2	EBOOL	● MP2	EBOOL
● IL3	EBOOL	● MP3	EBOOL
● IL4	EBOOL	● MP4	EBOOL
● IL5	EBOOL	● MP5	EBOOL
● IL6	EBOOL	● MP6	EBOOL
		● MP7	EBOOL

Figura 27: Variables de salida del sistema

A la hora de la simulación es importante que la dirección que hemos establecido a las variables de entrada y salida dentro del rack y de los módulos digitales (entradas/salidas) correspondientes sea la correcta y esté realizada con la nomenclatura correspondiente, también, hay que tener en cuenta si existe más de un bastidor, en nuestro caso, sólo utilizaremos uno. En la figura 28, vemos un ejemplo de cómo es la nomenclatura y lo que significa cada elemento.

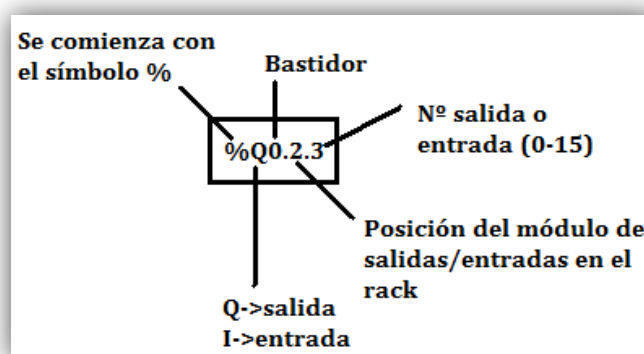


Figura 28: Nomenclatura de la dirección de las variables

Una vez que hemos introducido las direcciones de nuestras variables, si pinchamos dentro del módulo de entradas o de salidas de nuestro autómata, veremos que en su interior aparecen configuradas en las diferentes posiciones las variables que hemos establecido para la simulación con anterioridad, un ejemplo de ello, lo podemos observar en la figura 29. Además, las variables pueden ser introducidas de manera más sencilla, introduciendo directamente el nombre en las posiciones de los módulos como se observa en la figura 29, de manera que la dirección se establece de forma automática.

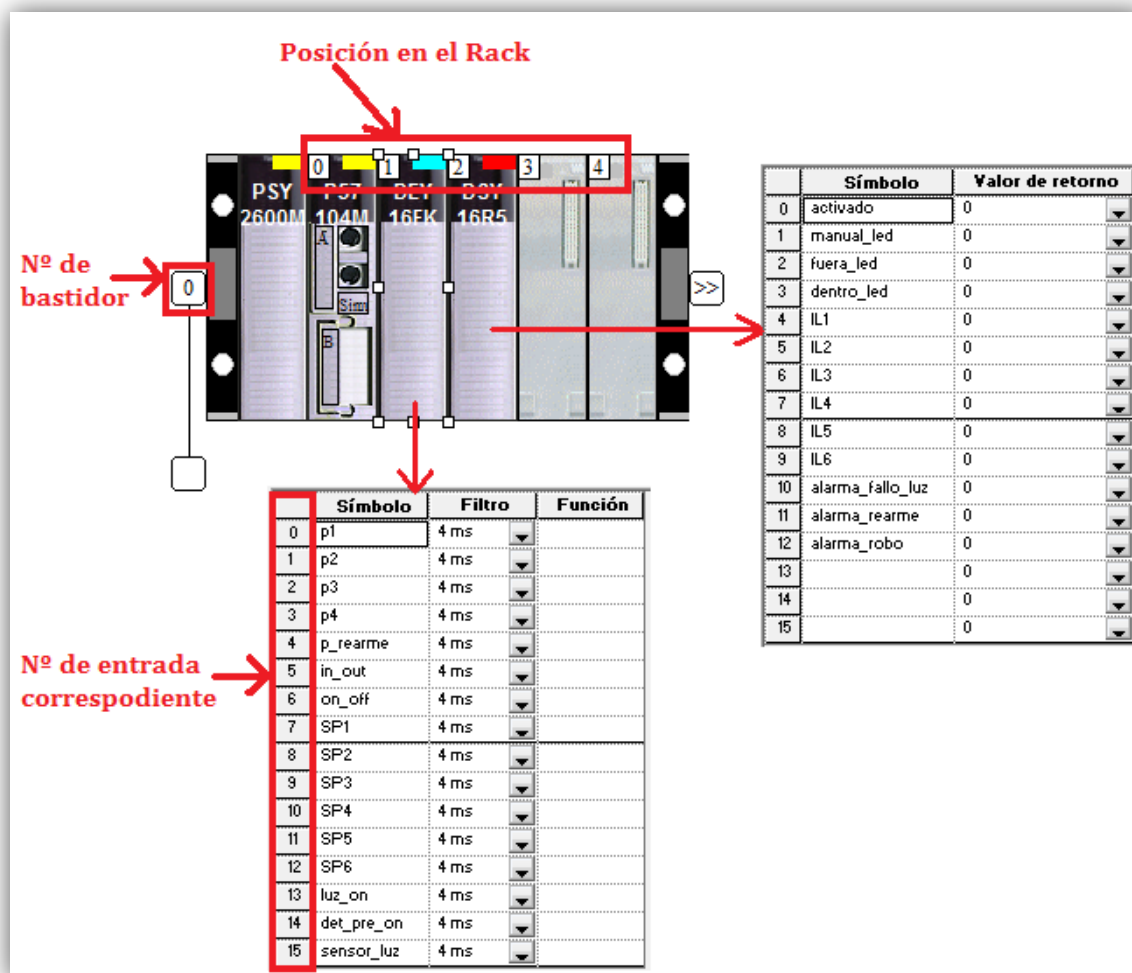


Figura 29: Comprobación e introducción de la dirección de las variables

#### 4.3.3. Programación de la solución en SFC

Una vez que hemos explicado las variables que vamos a utilizar en nuestro programa, se explicará cómo las usaremos y qué función desempeñará cada una dentro de él. En este apartado, se explicará la programación realizada tanto en el sistema principal (SFC principal), incluyendo sus etapas y transiciones, como en cada macroetapa (subsistemas), con el objetivo de comprender el funcionamiento del sistema a la perfección.

En primer lugar, se explicará el funcionamiento del SFC principal para poco a poco ir adentrándonos en la explicación de cada subsistema en detalle.



#### 4.3.3.1. SFC principal

El SFC principal que se puede observar en la figura 30, tiene en su interior varias macroetapas, éstas se explicarán cada una de manera individual. Este apartado, pretende ilustrar cómo se ha realizado la programación del sistema principal, entender su funcionamiento y saber cómo moverse dentro de él. Cabe mencionar que las transiciones en las que aparece un “1”, son aquellas que se cumplen siempre, o, en las cuales, no es necesaria una condición específica para pasar a la siguiente etapa, por ello tienen este valor.



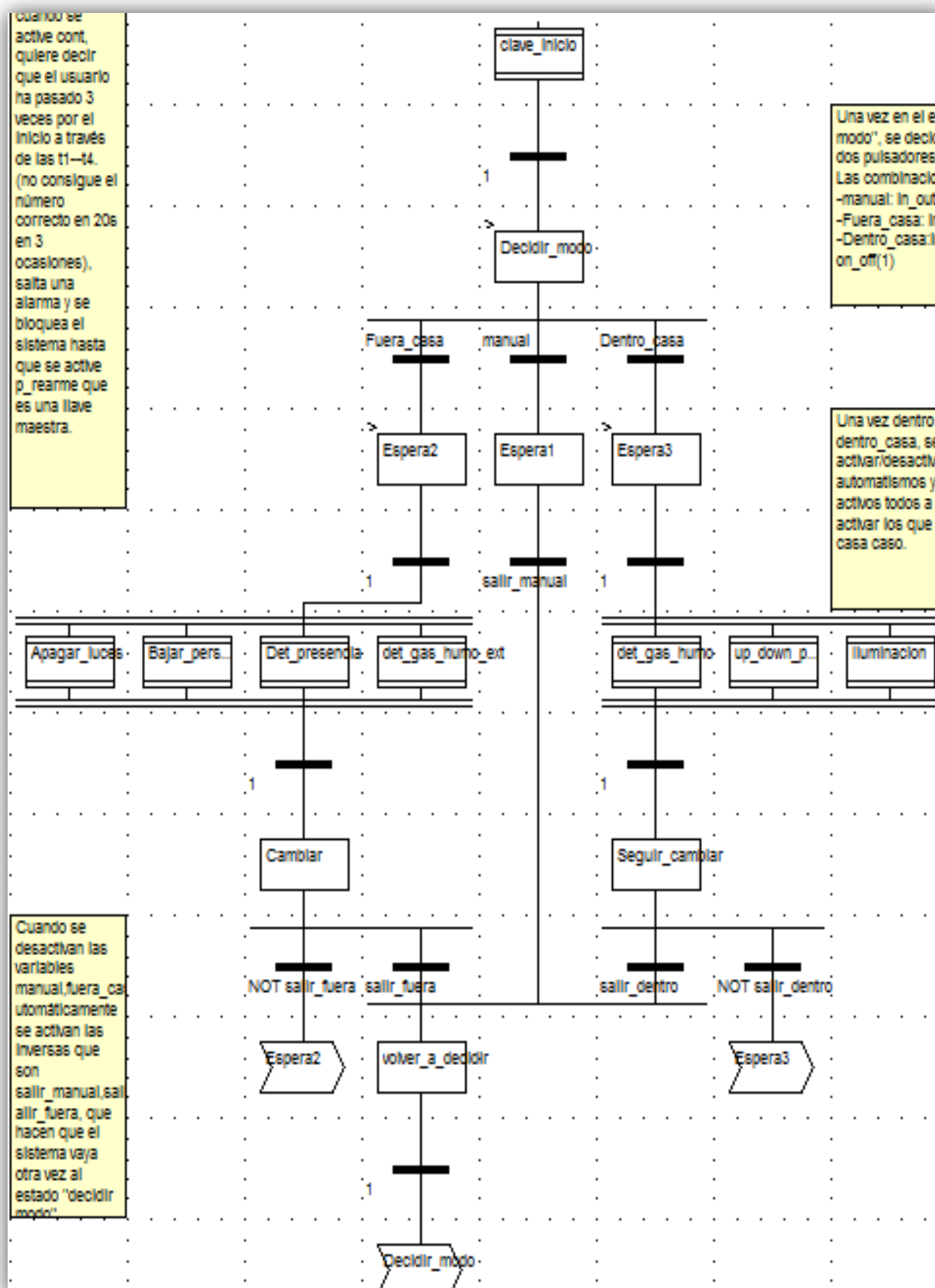


Figura 30: SFC principal

Una vez vista la estructura del SFC, explicaremos en detalle para qué sirve cada etapa y transición, y su programación interna realizada en LD:

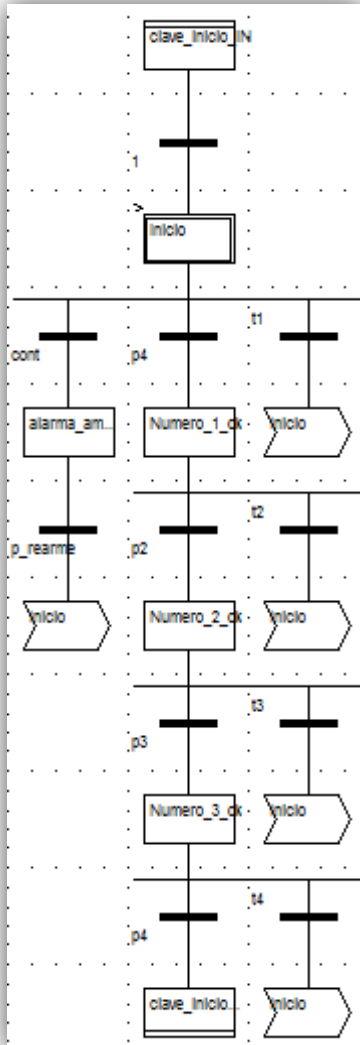


Figura 31: SFC clave seguridad

**Macroetapa “clave\_inicio”:** Implementa una clave de seguridad a la hora de activar el sistema mediante la introducción de una combinación de 4 dígitos, el SFC interno está estructurado como vemos en la figura 31. En el SFC se observa que al ser la primera transición un “1”, directamente nos situamos en la etapa inicio.

- Etapa inicio: Incorpora un contador que cuenta 3 errores, si se llega a este valor, se activará la variable “cont” y se pasará al estado “alarma\_amarilla”. El contador sólo aumenta la cuenta si detecta un flanco positivo de las variables t1, t2, t3 ó t4, y se resetea cuando se pulsa “p\_rearme”. También, si se detecta un flanco positivo en la variable “p\_rearme” se desactiva la variable “alarma\_rearme”, posteriormente veremos para qué sirve.

- Bifurcación alternativa cont/número 1 ok/t1: Aquí se plantean 3 caminos por los que poder seguir:

1. Camino “cont”: Al activarse la variable “cont” se pasa directamente a la etapa “alarma\_amarilla”, ésta representa una alarma en la introducción de la clave numérica de 4 dígitos, por lo tanto, al entrar en este estado se activará la variable “alarma\_rearme” que avisará de un fallo en el sistema. Se volverá a la etapa “inicio” cuando se pulse “p\_rearme”, y se desactivará la alarma como hemos explicado en la etapa inicial.
2. Camino “número 1 ok”: Pasaremos a esta etapa si encontrándonos en “inicio” pulsamos el primer número correctamente, en este caso “p4”. Las etapas posteriores “número\_2\_ok”, “número\_3\_ok” y “número\_4\_ok” funcionan de la misma manera pero pulsando los números correctos en cada caso.

3. Camino “t1”: La variable t1 se activará por dos motivos, o han pasado 20 sg en “inicio” y no se ha pulsado ningún número, o el número es incorrecto y se vuelve a “inicio” y se suma un error.

En el caso del camino “t1”, además de pulsar un número incorrecto al introducir la clave de seguridad, es necesario que hayan pasado al menos 3 sg en la etapa correspondiente, por ejemplo, en el caso del primer número, si se activa cualquier otro que no sea el “p4”, y, además, han pasado 3 sg en la etapa “inicio”, se activará la variable “t1” (esto último, se implementa mediante otro bloque de comparación y se utiliza para controlar la velocidad de ejecución del programa y que no sea demasiado rápida, lo que puede llegar a ocasionar problemas).

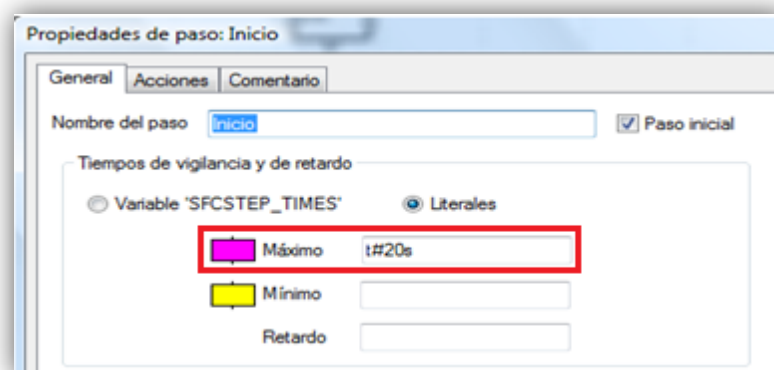


Figura 32: Introducción de tiempo de supervisión

El tiempo ha sido controlado mediante tiempos de supervisión en las etapas, esto quiere decir, que se usa una variable interna de cada etapa (la cual, controla el tiempo desde que se activa), evitando el uso de un temporizador específico. De esta manera, si llevamos más de 20 sg en “inicio”, la variable “Inicio.tmaxErr” se pone a uno, entonces, mediante un bloque comparador comprobamos dicha condición y activamos “t1” cuando se cumpla (ver anexo de programación LD). Esto se lleva a cabo incluyendo un tiempo máximo determinado en la etapa “inicio” como vemos en la figura 32.

El funcionamiento de las secciones de transición t2, t3 y t4 se realiza de igual manera, amoldando en cada caso los bloques compare a la etapa que cuenta el tiempo, y activando los pulsadores correspondientes, por ejemplo, en el caso de t2, en los bloques compare aparece la etapa “número\_1\_ok” y en los números que no debe pulsar los “p1”, “p3” y “p4”, por lo que el correcto es “p2” (ver anexo de programación LD).

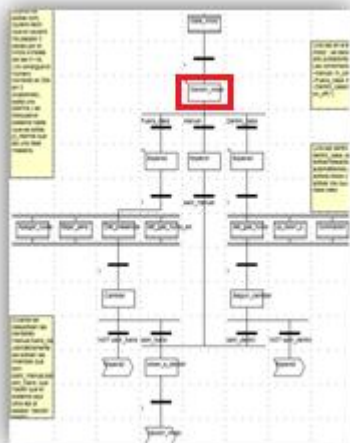


Figura 33: Detalle SFC principal 1

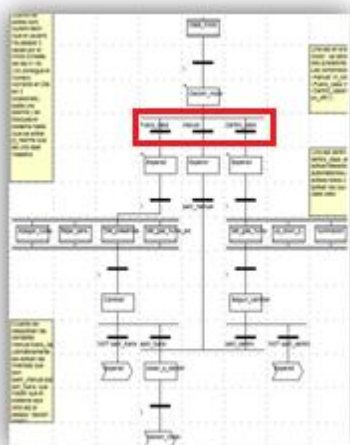


Figura 34: Detalle SFC principal 2

**Bifurcación alternativa “Fuera\_casa/manual/ Dentro\_casa”:** Aquí el camino se bifurca en 3 modos de funcionamiento diferentes, sólo estará activo 1 de los 3, aunque se podrá conmutar entre ellos como posteriormente explicaremos. Para elegir el modo deseado se han diseñado los botones “in/out” y “on/off”, según las combinaciones que vemos en la tabla 2, se activará un estado u otro:

MODOS	IN/OUT*	ON/OFF*
MANUAL	0	0
MANUAL	0	1
FUERA CASA	1	0
DENTRO CASA	1	1

Tabla 2: Código binario de conmutación entre modos (\*1-&gt;botón pulsado; 0-&gt; sin pulsar)

Cuando hayan pasado 20 sg, se comprobarán las combinaciones y se activará la transición correspondiente. Cabe destacar, que los bloques compare realizan la misma función que en “clave\_inicio”, están constantemente comprobando si “Decidir\_modos.tmaxErr=1” (que significa que han pasado los 20 sg).

Como sabemos y vemos en la figura 30 (SFC principal), el programa se bifurca en 3 caminos que funcionan individualmente entre sí, veremos cada uno de manera independiente:

**Camino “Manual”:** Elegido este modo se pasa al estado “Espera1”, aquí se activa la variable “manual\_led” que indica que el modo manual se encuentra activo, se saldrá de este modo y se volverá a “Decidir\_modos” cuando se detecte que ha cambiado la combinación de botones “in/out” y “on/off” que lo activa. Al salir se activa la variable “salir\_manual”, la cual, está incluida en la conjunción alternativa que se ve en la figura 35, y que retorna a “Decidir\_modos”, además, se pone a cero “manual\_led”, ya que el modo manual ya no se encuentra activo.

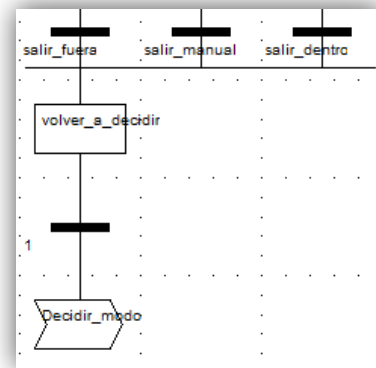


Figura 35: Detalle conjunción alternativa SFC principal

**Camino “Fuera\_casa”:** En la etapa “Espera2” igual que en “Espera1”, se activa un led (“fuera\_led”=1) que indica que el modo fuera está activo, sin embargo, a diferencia del caso anterior, en este no se sale directamente a la conjunción alternativa de salida (figura 35), sino que se pasa antes por algunas etapas previas que ahora explicaremos.

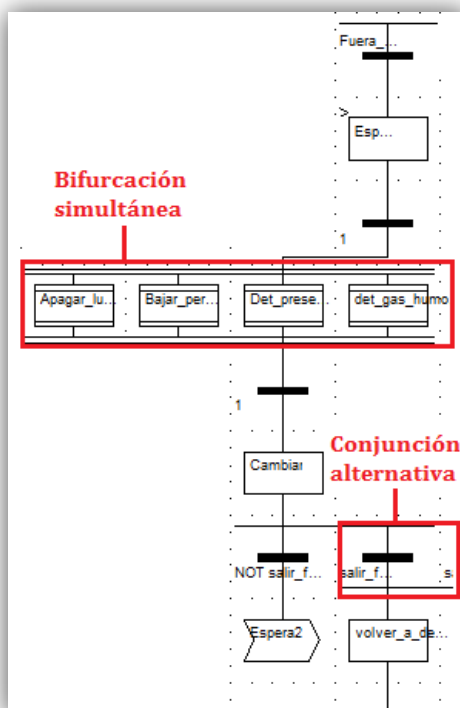


Figura 36: Detalle camino fuera\_casa SFC principal

**Bifurcación simultánea:** Si observamos la figura 36, vemos que si pasamos del estado “Espera2” nos encontramos con una bifurcación simultánea de 4 caminos que se realizarán a la vez (apagar luces, bajar persianas, conectar alarma de presencia, y de gas y humo), únicamente cuando todos estos caminos hayan terminado correctamente su función, pasaremos a la etapa “Cambiar”. Las macroetapas se explicarán después del funcionamiento de cada modo en un apartado específico por su extensión y complejidad.

**Etapas “Cambiar”:** Cuando no se cumpla la combinación de los botones “in/out” y “on/off” que activa el modo fuera de casa, la variable “salir\_fuera” se activará (y se apaga “fuera\_led”), si esto sucede se saltará a “Decidir\_modos”, pero si “salir\_fuera” se encuentra desactivada (no se han variado los botones “in/out” y “on/off”), el programa saltará a “Espera2” y realizará un nuevo ciclo de ejecución (ver figura 36).

**Camino “Dentro\_casa”:** Este modo funciona igual que el anterior, en la etapa “Espera3” se activa un led por la puesta a 1 de la variable “Dentro\_led” y automáticamente se pasa de nuevo a una bifurcación simultánea.

Bifurcación simultánea: Aquí se realizan 3 caminos de manera simultánea, la subida o bajada de persianas, el encendido o apagado de la luz y la activación de la alarma de gas y humo, una vez que todas hayan finalizado su labor, se pasará al estado “Seguir\_cambiar”. En la figura 37, podemos ver el diagrama secuencial del modo dentro de casa.

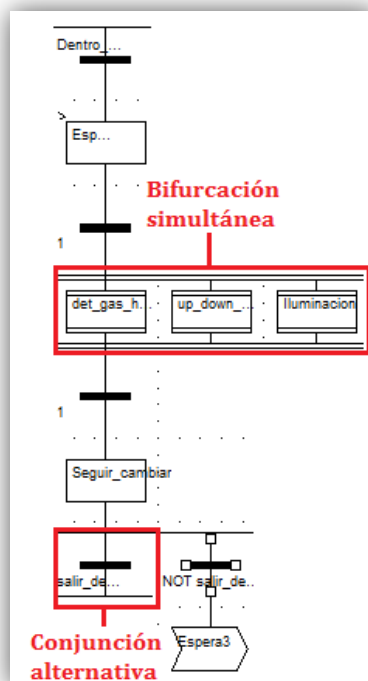


Figura 37: Detalle camino dentro\_casa SFC principal

Etapa “Seguir cambiar”: Funciona igual que la etapa “Cambiar” del modo fuera de casa, se comprueba si el usuario quiere conmutar de modo, si es así, se activa “salir\_dentro” y se pasa a “Decidir\_modos”, si no, se vuelve a “Espera3” para volver a ejecutar un nuevo ciclo interno del modo.

#### 4.3.3.2. Macroetapas del modo fuera de casa

1. **Apagar\_luces:** Este sistema como se ha comentado en el capítulo 4.2, apaga las luces antes de salir de casa de manera autónoma.

Etapa “comprobar luces apagadas ext”: Lo primero que se hace es resetear todos los actuadores que activan las luces en cada habitación, posteriormente, se comprueba que todos ellos (“IL1-IL6”) estén apagados, de ser así, se activaría la variable “ok\_luces” y se saldría de la macroetapa. Si por el contrario, pasan 60 sg y no se han conseguido apagar todas las luces, se activará la variable “alarma\_fallo\_luz” y pasaremos a la siguiente etapa como podemos ver en la figura 38.

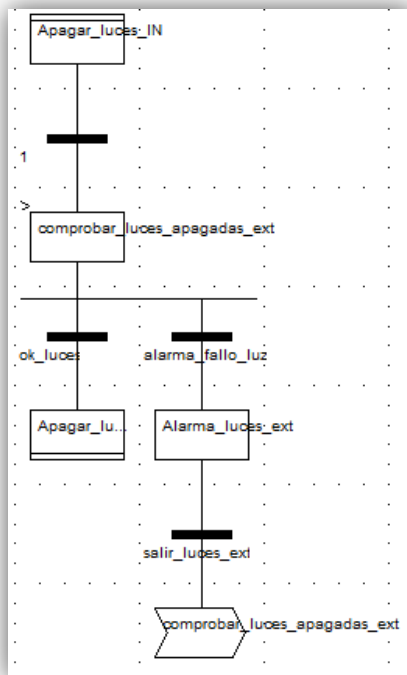


Figura 38: Macroetapa apagar\_luces modo fuera de casa

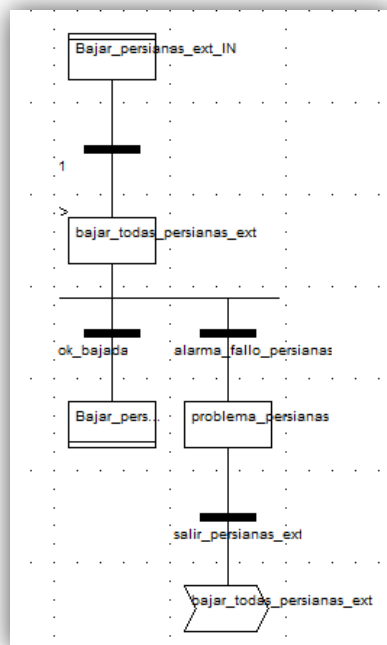


Figura 39: Macroetapa bajar\_persianas modo fuera de casa

**Etapa “Alarma luces ext”:** Se trata de un estado de alarma que activa la variable “led\_luz”, ésta se usa para representar un fallo producido al apagar las luces. El led se apagará cuando el usuario restablezca el funcionamiento normal del sistema pulsando “p\_rearme” (a su vez, se activará la transición “salir\_luces\_ext”), volviendo a la etapa “comprobar-luces-apagadas-ext” para realizar otro ciclo de ejecución. En realidad, el programa nunca llegará a entrar en este estado de alarma, debido a que se realiza una orden de apagado de luces general con anterioridad a él, la cual, no fallará nunca, sin embargo, se ha realizado como medida de seguridad ante una posterior implementación en un sistema real que incorpore un programa más exhaustivo de comprobación de activación de la luz.

2. **Bajar\_persianas\_ext:** Procede a bajar las persianas de manera automática.

**Etapa “Bajar todas persianas ext”:** Se comprueba que todos los sensores final de carrera (FC1-FC7) se encuentren en la posición o (a nivel bajo), si alguno de ellos se encuentra a 1, automáticamente se activa el motor correspondiente (MP1-MP7) para cambiar su posición; una vez que todas las persianas se encuentren bajadas, se activa “ok\_bajada” y se sale de la macroetapa. Si por alguna razón, pasan 60 sg y no se ha realizado la bajada de las persianas (no se encuentran todos los sensores final de carrera a nivel bajo), se activa la transición “alarma\_fallo\_persianas” para avisar al usuario de este suceso, y se pasa al estado “problema\_persianas”.

Etapa “problema persianas”: En esta etapa se procede a encender una alarma luminosa (“led\_error\_persianas”) para avisar al usuario de una anomalía en el sistema. Para regresar a un funcionamiento normal y apagar la alarma, el usuario tiene que pulsar “p\_rearme” y el sistema volverá a la etapa “Bajar\_todas\_persianas\_ext”.

3. **Det\_presencia**: Activa el detector de presencia en el interior de la casa. Este sistema se puede habilitar o deshabilitar con el interruptor “det\_pre\_on”, si el interruptor está apagado, se saldrá directamente de la macroetapa, ya que el sistema no se encuentra activo, si por el contrario, “det\_pre\_on” se encuentra pulsado, pasaremos a la etapa “Sensores\_presencia”.

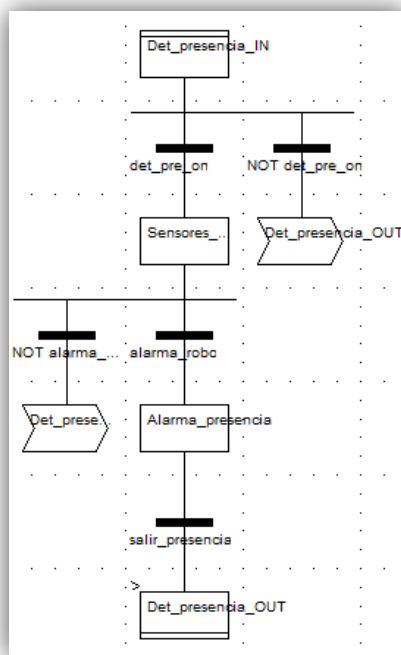


Figura 40: Macroetapa  
det\_presencia modo fuera de casa

Etapa “Sensores presencia”: Los sensores de presencia (SP1-SP6) utilizados para encender y apagar las luces en el modo dentro de casa son reutilizados para un uso de seguridad en este subsistema, aquí se comprueba constantemente la señal recibida y se activa la transición “alarma\_robo” en el caso de detectar presencia, si no es así, se sale de la macroetapa.

Etapa “Alarma presencia”: Al entrar en este estado, automáticamente se activa la alarma de robo (“led\_robo”) para avisar al propietario de una intrusión en su vivienda, al igual que en los demás sistemas de alarma, el desbloqueo se produce con la pulsación de “p\_rearme” (y se apaga “led\_robo”).

4. **Det\_gas\_humo\_ext**: Este sistema se ocupa de vigilar la casa ante posibles incendios o fugas de gas, e intentar evitar explosiones y posteriores daños derivados de estas situaciones de peligro. Al igual que el detector de presencia tiene un interruptor que habilita su uso “det\_gh\_on” y, sin el cual, la macroetapa estaría desactivada.



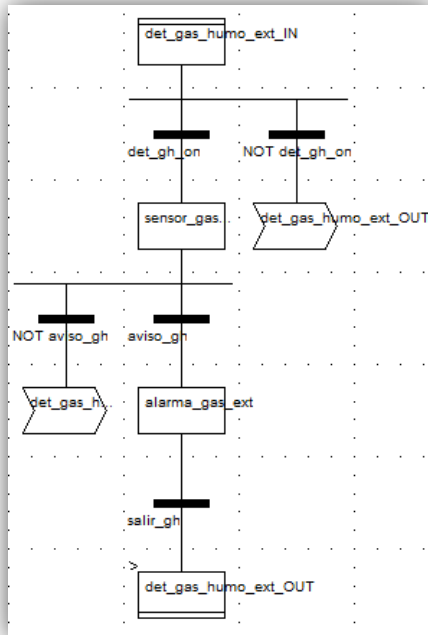


Figura 41: Macroetapa  
det\_gas\_y\_humo modo fuera de casa

y anexos de programación LD).

Etapa “sensor gas ext”: Se supervisan constantemente un sensor de humo (“sensor\_humo”) y un sensor de gas (“sensor\_gas”), de manera que si en algún momento cualquiera de los dos se activa, se procede a cerrar la electroválvula de gas (“EV\_gas”) para evitar una explosión y se activa la transición “aviso\_gh”. En el caso de que los sensores no detecten gas ni humo, se sale de la macroetapa como podemos ver en la figura 41.

Etapa “alarma gas ext”: Se comporta igual que todos los estados de alarma implementados hasta ahora, activa automáticamente la alarma de gas y humo “led\_gh” hasta que detecta que se ha pulsado “p\_rearme”, es entonces, cuando apaga la alarma “led\_gh”, abre de nuevo la electroválvula de gas (“EV\_gas”) y vuelve al estado normal de funcionamiento (ver figura 41

#### 4.3.3.3. Macroetapas del modo dentro de casa

1. **Det\_gas\_humo:** Es idéntico al sistema que se utiliza en el modo fuera de casa, la única diferencia radica, en que los estados de cada macroetapa se llaman de manera diferente, pero las variables, transiciones y acciones programadas son las mismas, por lo que no volveremos a repetir la explicación.
2. **Up\_down\_persianas:** Este subsistema sirve para subir o bajar las persianas de la vivienda automáticamente teniendo en cuenta las condiciones de luminosidad exterior, además, se puede deshabilitar su uso mediante el interruptor de encendido “per\_on”, proporcionando al sistema de total flexibilidad en su uso.

Primero, se comprueba si el botón “per\_on” está habilitado, si es así, se pasará a la etapa “iluminacion\_ext”.

Etapa “Iluminación ext”: Se tiene en cuenta la señal que se recibe del sensor de iluminación colocado en el exterior de la vivienda (“sensor\_luz”), si éste no se activa, significa que hay poca luz y hay que bajar las persianas, si no es así, habrá que subirlas.

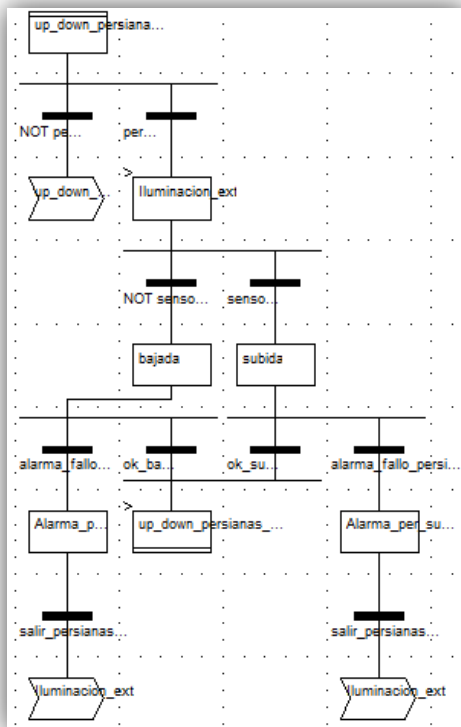


Figura 42: Macroetapa up\_down\_persianas modo dentro de casa

**Etapa “subida”:** La etapa subida funciona de manera análoga que la de “bajada”, la diferencia radica, en que ahora se comprueba que los sensores final de carrera nos devuelvan el valor 1 (posición arriba de la persiana), por lo demás, funciona de igual manera. El estado de alarma “alarma\_per\_subida” comparte la misma acción programada que “alarma\_per\_bajada”, por lo que el led que se enciende al detectar un error en ambos funcionamientos es el mismo (“led\_error\_persianas”).

3. **Iluminación:** Igual que los demás sistemas incorporados en este modo de funcionamiento, la iluminación contiene un interruptor de encendido/apagado (“luz\_on”), si no se encuentra activo, se salta la macroetapa, y si lo está, se pasa a la etapa “sensor-iluminacion”.

**Etapa “sensor iluminacion”:** Simplemente se tiene en cuenta el estado del sensor de luz exterior, si éste se encuentra a 0, se cogerá el camino “todo\_con\_luz”, debido a que la iluminación exterior no es suficiente (es de noche, hay nubes, tormenta etc...), y si el sensor detecta luz, se pasará a “bano\_garaje”.

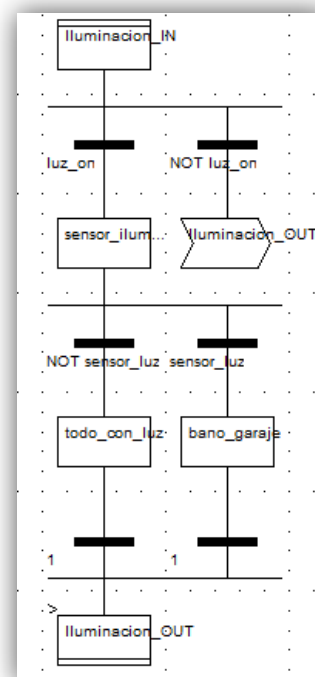


Figura 43: Macroetapa iluminación modo dentro de casa

Etapla “todo con luz”: En esta etapa se encuentran activas todas las luces de la vivienda, pero sólo se encenderán cuando se detecte presencia en las habitaciones mediante los sensores de presencia (“SP1-SP6”), los cuales, activarán las luces correspondientes a cada habitación (“IL1-IL6”). Las luces permanecerán encendidas mientras se detecte presencia con los sensores, cada habitación lleva asociado un temporizador, de manera que si pasan 10 sg y no se ha detectado movimiento, se procederá a apagar la luz correspondiente en cada caso. La luz se volverá a encender automáticamente cuando se vuelva a detectar presencia.

Etapla “bano garaje”: Sólo se encuentran activos el baño y el garaje por ser las zonas oscuras de la vivienda y necesitar de iluminación artificial a pesar de haber luz exterior suficiente, por lo demás, el funcionamiento es igual a la etapa anteriormente explicada, aunque en este caso, estarán activos únicamente los sensores de presencia SP1 y SP2, y sólo se encenderán las luces IL1 e IL2 correspondientes al baño y al garaje.

#### 4.4. Creación de tablas de variables

Como se comentó en capítulos anteriores, una de las maneras de simular el sistema es la realización de tablas de animación, este apartado se centrará en explicar qué tablas se han creado y su uso dentro del programa.

Durante la realización del proyecto se han creado varias tablas de animación que no serán usadas en la simulación del sistema global, pero que sin embargo, se han conservado en el archivo final, esto se debe, a que la solución del problema se ha realizado de manera escalonada, me explico, primero se realizó el sistema clave de entrada, y una vez que se comprobó que funcionaba (mediante una tabla de animación específica), se unió a una bifurcación alternativa para conmutar de modos (con otra tabla para su funcionamiento), y así, hasta conseguir el simulador completo. En la figura 43, podemos ver las diferentes tablas que se han implementado y un detalle de la creada para la comprobación de la clave de entrada.

Nombre	Valor	Tipo
alarma_reame		EBOOL
cont		BOOL
in_out		EBOOL
on_off		EBOOL
p1		EBOOL
p2		EBOOL
p3		EBOOL
p4		EBOOL
p_reame		EBOOL
t1		BOOL
t2		BOOL
t3		BOOL
t4		BOOL
Dentro_casa		BOOL
manual		BOOL
Fuera_casa		BOOL

Figura 44: Tablas de animación

A pesar de todas las tablas creadas sólo se han usado dos tablas para la simulación del sistema domótico, una para las entradas y otra para las salidas. Se distinguió entre entradas y salidas, con el objetivo de separar la información, y con ello facilitar la simulación. En las figuras 45 y 46, podemos ver imágenes de ambas tablas de animación:

Nombre	Valor	Tipo	Comentario
det_gh_on		EBOOL	Interruptor para on/off el sistema de alarma gas/humo
det_pre_on		EBOOL	Interruptor para on/off el sistema de alarma presencia
FC1		EBOOL	Sensor final carrera 1, los finales de carrera a 1 indican persiana subida//estudio1
FC2		EBOOL	Sensor final carrera 2//garaje
FC3		EBOOL	Sensor final carrera 3//cocina
FC4		EBOOL	Sensor final carrera 4//pasillo
FC5		EBOOL	Sensor final carrera 5//salon
FC6		EBOOL	Sensor final carrera 6//estudio2
FC7		EBOOL	Sensor final carrera 7//cocina2
in_out		EBOOL	Interruptor dentro/fuera, si está pulsado es dentro, si no fuera.
luz_on		EBOOL	Interruptor que activa el funcionamiento automático de las luces
on_off		EBOOL	Interruptor encendido/apagado, si está pulsado esta encendido lo automático si no no.
p1		EBOOL	Pulsador 1 para clave numérica entrada
p2		EBOOL	Pulsador 2 para clave numérica entrada
p3		EBOOL	Pulsador 3 para clave numérica entrada
p4		EBOOL	Pulsador 4 para clave numérica entrada
p_reame		EBOOL	Pulsador de rearme para todos los sistemas de alarma.
per_on		EBOOL	Interruptor que enciende/apaga las persianas automáticas
sensor_gas		EBOOL	Sensor que detecta fuga de gas
sensor_humo		EBOOL	Sensor que detecta fuga de humo
sensor_luz		EBOOL	Sensor que detecta la luminosidad exterior, 0= no hay luz suficiente y 1= hay luz natural
SP1		EBOOL	Sensor presencia 1 (baño)
SP2		EBOOL	Sensor presencia (garaje)
SP3		EBOOL	Sensor presencia 3 (cocina)
SP4		EBOOL	Sensor presencia 4 (pasillo)
SP5		EBOOL	Sensor presencia 5 (salon)
SP6		EBOOL	Sensor presencia 6 (estudio)

Figura 45: Tabla de animación “entradas”












Modificación    Forzar           			
Nombre	Valor	Tipo	Comentario
● activado		EBOOL	Sistema Activo
● alarma_reame		EBOOL	Sistema Bloqueado
● dentro_led		EBOOL	Led que avisa de modo dentro de casa
● EV_gas		EBOOL	Interruptor que cierra la electroválvula del gas cuando esta a 1
● fuera_led		EBOOL	Led que avisa de modo fuera de casa
● IL1		EBOOL	Interruptor luz 1 (baño)
● IL2		EBOOL	Interruptor luz 2 (garaje)
● IL3		EBOOL	Interruptor luz 3 (cocina)
● IL4		EBOOL	Interruptor luz 4 (pasillo)
● IL5		EBOOL	Interruptor luz 5 (salon)
● IL6		EBOOL	Interruptor luz 6 (estudio)
● led_error_persianas		EBOOL	Led que indica que en el modo fuera de casa alguna persiana no funciona correctamente (no se ha podido bajar)
● led_gh		EBOOL	Alarma que avisa de una fuga de gas o incendio
● led_luz		EBOOL	led de alarma de que quedan luces encendidas que no se han podido apagar automáticamente en el modo salir
● led_robo		EBOOL	Alarma que avisa de intrusos
● manual_led		EBOOL	Led que avisa de modo manual de funcionamiento
● MP1		EBOOL	Motor persiana 1
● MP2		EBOOL	Motor persiana 2
● MP3		EBOOL	Motor persiana 3
● MP4		EBOOL	Motor persiana 4
● MP5		EBOOL	Motor persiana 5
● MP6		EBOOL	Motor persiana 6
● MP7		EBOOL	Motor persiana 7

Figura 46: Tabla de animación “salidas”

En el Capítulo 5 se explicará cómo se realiza la simulación utilizando este tipo de tablas y veremos un ejemplo detallado.

#### 4.5. Diseño de la pantalla de operador

Como se comentó en el apartado 2.3 del Capítulo 2, un simulador debe ser una imitación fiel del sistema real que posteriormente será utilizado, con el fin de poder anticipar situaciones futuras y que el alumno pueda estar preparado ante ellas. Además, debe cumplir varios requisitos como, por ejemplo, ser intuitivo, de fácil de manejo y entendimiento, y sobretodo, útil para la labor que ha sido creado.

El simulador o pantalla de operador ha sido creado teniendo en cuenta muchos de estos aspectos clave, los cuales, son imprescindibles para su buen entendimiento y funcionamiento. En nuestro caso de estudio (vivienda automática), el simulador debe cumplir varios requisitos adicionales:

- Se deben poder observar los cambios producidos en cada habitación, por ejemplo, subida de persianas, encendida de luces, alarmas, etc...
- Cuando se produzca un estado de alarma debe ser fácilmente observable a simple vista (debe llamar la atención).
- La botonera debe estar bien diferenciada del resto de partes del simulador.
- Si el sistema tiene gran cantidad de entradas y salidas, se deberá poner una leyenda para facilitar el entendimiento de los iconos, imágenes, etc...
- Se deben observar con facilidad los cambios de estado del sistema, por ejemplo, paso del modo manual al modo fuera de casa.

#### 4.5.1. Funcionamiento de la pantalla de operador

En la figura 47, vemos la estructura general del simulador domótico, se puede observar que es un sistema interactivo, el cual, se podría implementar en una pantalla táctil a la entrada de la casa, en un ordenador, tablet, Smartphone, etc... El simulador se diseñó con la idea de que el usuario pudiera tener el control de los sistemas desde cualquier sitio de la casa y, además, supiera en todo momento el estado de ellos. En este apartado, veremos cómo hemos estructurado el simulador (los elementos que se han usado y la manera de incluirlos) y se explicará su funcionamiento.

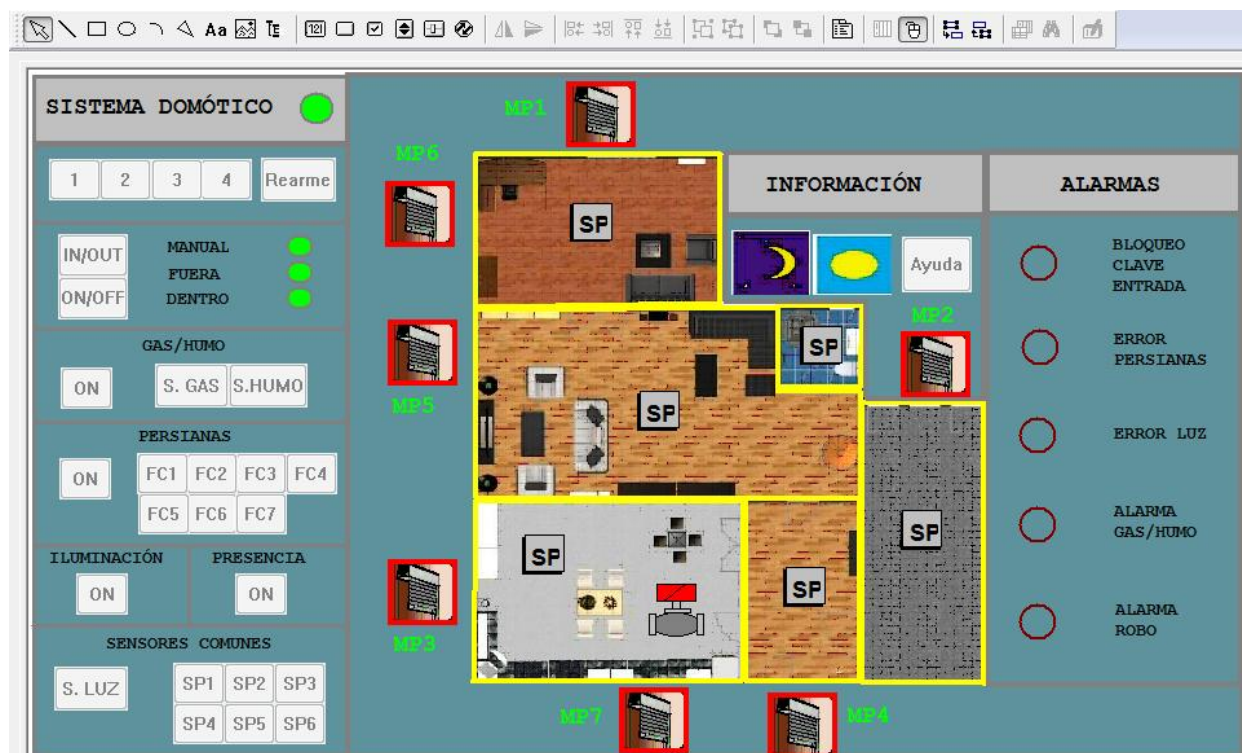


Figura 47: Pantalla principal del simulador domótico



Si observamos la figura 47, vemos que en la parte izquierda se sitúa la botonera, los botones se encuentran repartidos en subsistemas (iluminación, gas/humo, presencia, etc...) para facilitar su uso, a excepción del botón de “AYUDA”. Éste se encuentra incorporado dentro del simulador, al pulsarlo, se abrirá automáticamente la pantalla de ayuda que observamos en la figura 48, donde aparece una leyenda con todas las imágenes e iconos utilizados en el simulador y su significado, además de una breve explicación del funcionamiento del sistema.

Una vez entendido el funcionamiento, para volver a la pantalla del simulador pinchamos en “Domótica” y, automáticamente, se abrirá nuestro simulador. Además de los botones, aparecen 4 leds verdes, éstos sólo aparecerán durante la simulación si se cumplen determinadas condiciones, por ejemplo, el primer led (el grande) se activará cuando se haya introducido correctamente la clave de entrada, los demás se encenderán cuando se active cada modo correspondiente (manual/dentro o fuera).

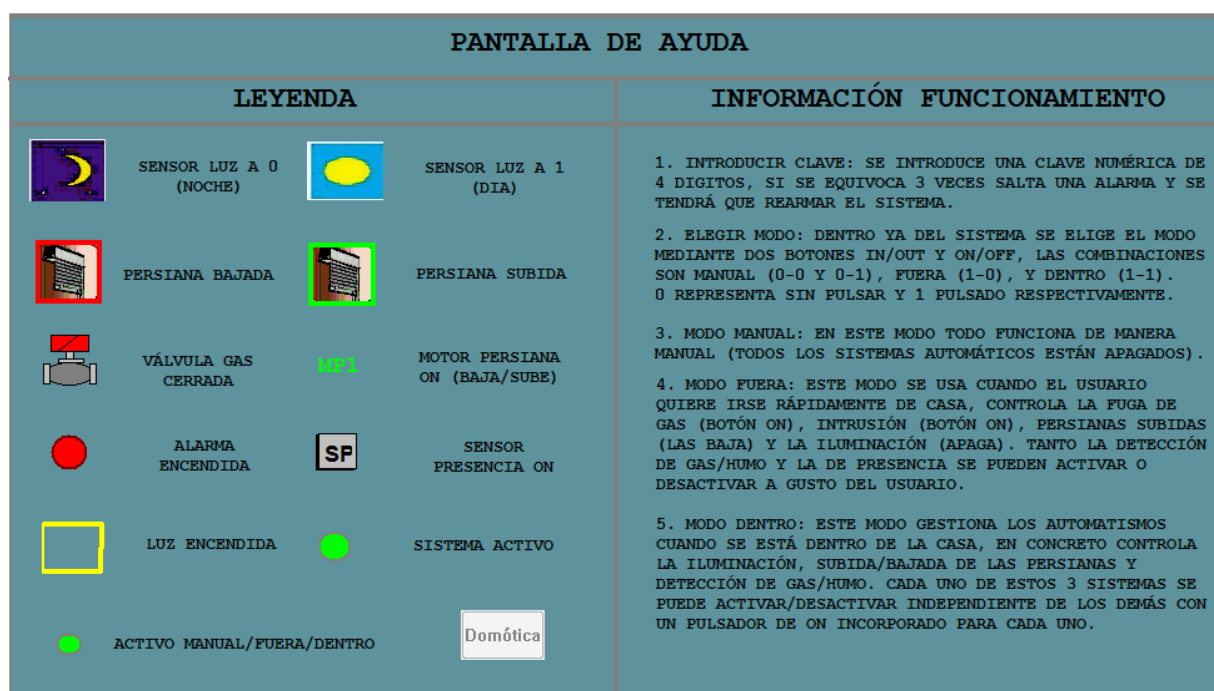


Figura 48: Pantalla de ayuda del simulador domótico

Si de nuevo observamos la figura 47, vemos que en su parte derecha existe un apartado “ALARMAS”, aquí se han juntado todos los leds de aviso, de manera que si se produce cualquiera de las 5 posibles alarmas parpadeará un led rojo para indicarlo. Por último, ya dentro de la zona de simulación (zona central figura 47) se aprecia otro apartado con el título “INFORMACIÓN”, aquí podemos ver el estado del sensor de iluminación exterior de manera muy intuitiva, siendo representado con una luna si se encuentra a “0” y con un sol si lo está a “1” (ver LEYENDA en figura 48). En esta zona, también se encuentra el botón “Ayuda” que redirecciona a la pantalla de ayuda.

Explicada la botonera, las alarmas y el apartado de información, hablaremos en detalle del cuerpo del simulador. La vivienda se representa mediante un plano de planta, en él se diferencian claramente los 6 habitáculos, lo cual, era uno de nuestros objetivos a priori. Como la resolución de la imagen no es muy elevada y las persianas no se distinguen a simple vista, se optó por colocar un icono justo al lado de cada una, el cual, hiciera saber de la existencia de una ventana en la pared de la habitación en cuestión (ver figura 47).

Según el estado de las persianas, aparecerá un cuadrado verde (subidas) o rojo (bajadas) alrededor de este icono, además, si se está produciendo un cambio de estado de una persiana, aparecerá junto al icono en verde MPX, siendo X el número de la persiana de 1-7, por ejemplo, la persiana 2 se encuentra bajada (cuadrado rojo) y se quiere subir, por lo que aparece en verde junto a ella MP2, entonces, se pulsará el botón FC2 (indica que el sensor final de carrera ha detectado la posición requerida) y, automáticamente, el recuadro rojo que rodea al icono se volverá verde.

La representación de la iluminación se realiza de forma similar, esta vez, aparecerá un recuadro amarillo alrededor de la habitación que se encuentre con las luces encendidas, además de esto, aparecerá un icono dentro de cada habitación que representará al sensor de presencia (ver figura 47), si este desaparece y pasan 10 sg, se apagará automáticamente la luz (se quitará el recuadro amarillo).

Para finalizar, sólo queda explicar la representación de las alarmas de gas/humo y presencia, ambas funcionan de manera muy semejante. La alarma de gas/humo será activada si pulsamos el sensor de humo o el de gas, en ese momento, aparecerá un led parpadeante en el apartado de alarmas y en la cocina un icono de una válvula cerrada, el cual, representa el cierre de la electroválvula de gas (ver figura 47). En la detección de presencia, al activarse el detector de presencia en cualquier habitación, parpadeará el led de alarma robo.



#### 4.5.2. Elementos usados en la pantalla de operador

En este apartado, explicaremos cómo se han implementado y configurado los diferentes iconos, imágenes, botones y formas utilizados para realizar la simulación, para ello, veamos primero la barra de herramientas que incorpora el entorno de diseño:

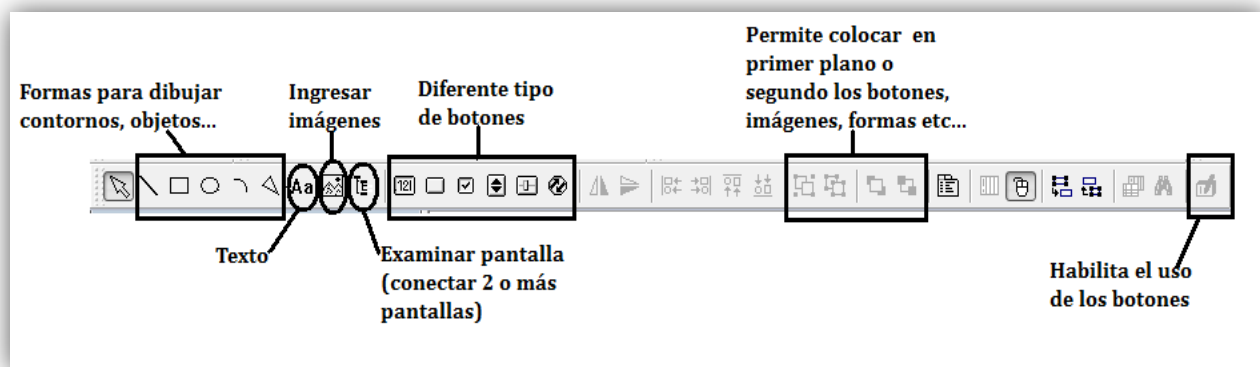


Figura 49: Barra de herramientas IOSEditor

Después de esta imagen autoexplicativa, veremos en detalle qué elementos dentro de los nombrados en la figura 49 se han usado dentro del simulador y de qué manera.

1. **Formas:** Se han usado para representar los leds, los cuadrados de estado de las persianas y el estado de las luces (cuadro amarillo alrededor de las habitaciones). Hay varias maneras de configurarlos, pueden aparecer fijos desde el inicio o pueden estar asociados al estado de una variable y aparecer con ella (bit=0, bit= 1, un determinado rango de valores...). También, al aparecer puede hacerlo de manera fija o parpadeante, en la figura 50, vemos un ejemplo de un led verde que se activa cuando la variable `manual_led=1` y que permanece fijo hasta que cambia el estado de dicha variable, además, se ven explicaciones de las diferentes pantallas donde se configuran estas condiciones.

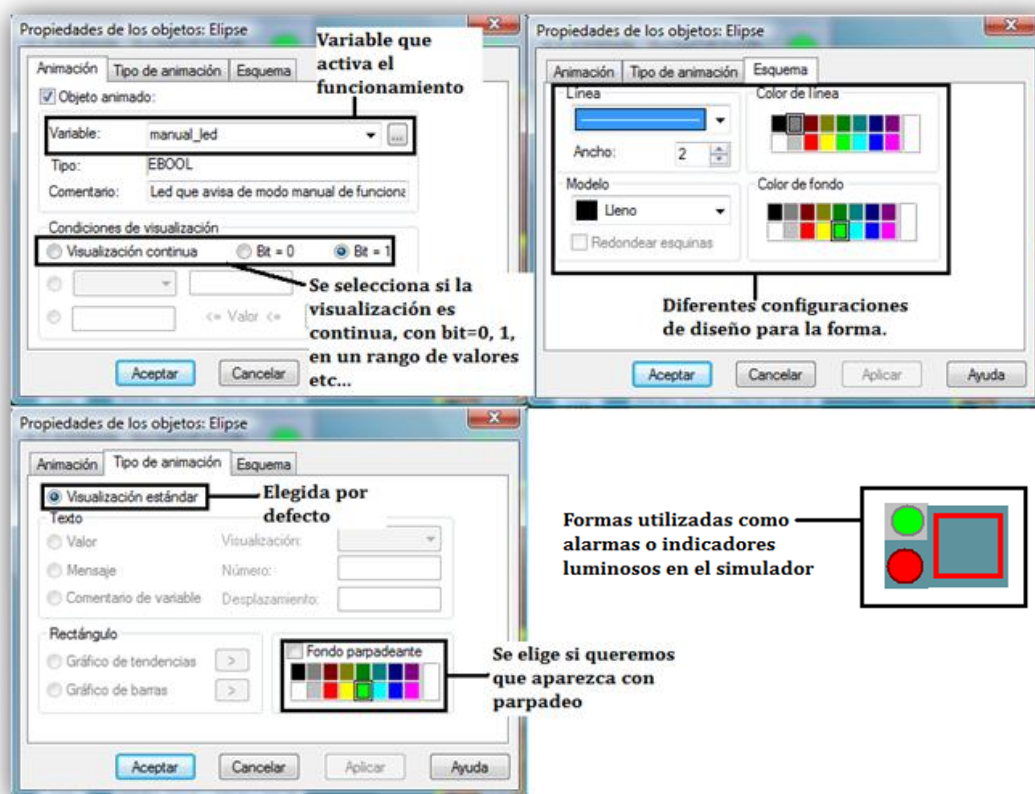


Figura 50: Ejemplo de configuración de forma

2. Texto: Además de formas, también podemos añadir texto, tanto de forma individual o como complemento a una imagen o forma, etc... El texto se puede configurar de la misma manera que las formas, se puede asociar al valor de una variable, puede anclarse de manera constante, puede aparecer de manera fija o parpadeante, la única diferencia es que tiene una pestaña para poder editar y personalizar el texto, en la figura 51 se puede observar un ejemplo.

Texto utilizado en el simulador de manera fija (modos) o asociado a variables (MP1).

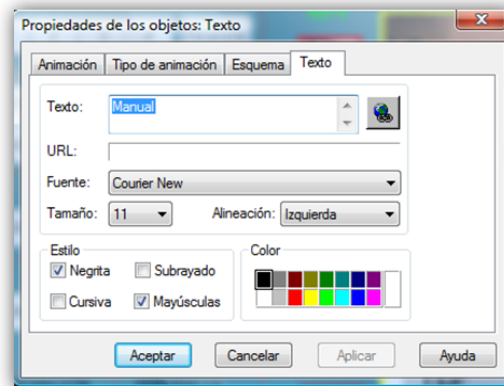
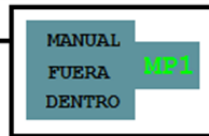


Figura 51: Ejemplo de configuración de texto

3. **Ingresar imágenes:** Una vez que hemos pulsado el icono de introducir imagen y la hemos seleccionado, tenemos que configurar su funcionamiento. Las imágenes también pueden ser asociadas a una variable y aparecer cuando ésta tome algún valor específico (sensor de luz), o formar parte del simulador de manera fija (como la imagen de planta de la vivienda). Por último, para que la imagen y su configuración aparezcan después de haber guardado el proyecto, debemos pulsar el botón “Imagen copiada en el espacio actual” (ver figura 52), el cual, creará una copia de nuestra imagen y su configuración para utilizarla en posteriores sesiones.

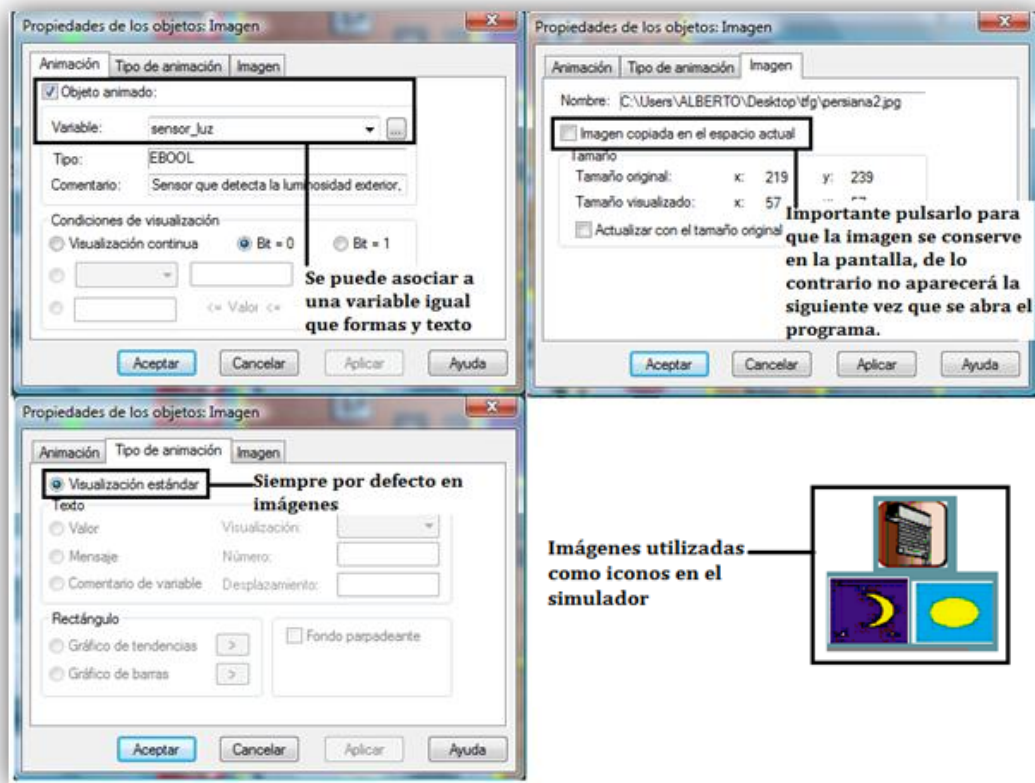


Figura 52: Ejemplo de configuración de imagen

4. Examinar pantalla: Este botón sirve para navegar entre varias pantallas, es muy útil en sistemas que tienen varias pantallas de operador, nosotros lo utilizamos para unir la pantalla principal del simulador a una pantalla de ayuda, la cual, contiene una leyenda e instrucciones básicas para usar el simulador. También, como en elementos anteriores, puede asociarse a una variable para que sea visible en determinado momento de la simulación o ser mostrada de manera continua. Para seleccionar la pantalla de destino deseada, se busca en valor como se observa en la figura 53, además, se puede asociar el nombre de la pantalla al botón o el valor de orden (adjudicado por orden de creación) que tiene dicha pantalla.

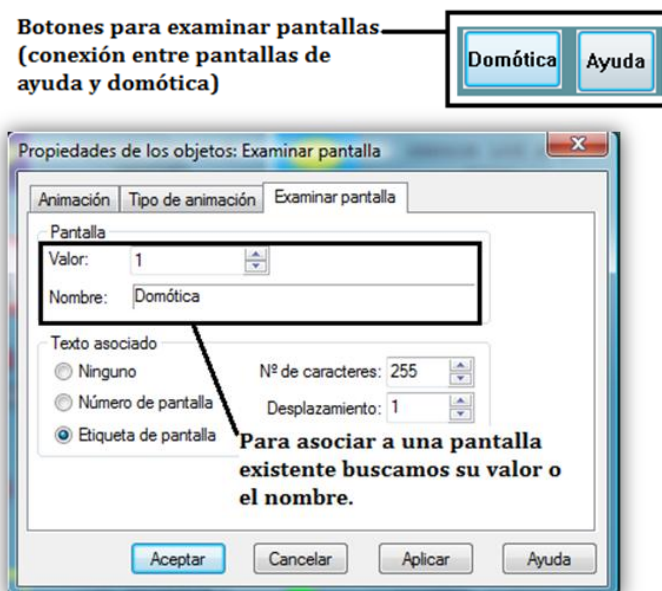


Figura 53: Ejemplo de configuración de botón examinar pantalla

5. Botón de comando: Lo que diferencia al botón de comando de los elementos explicados con anterioridad es que puede controlar el valor de una determinada variable para influir en el funcionamiento del sistema, puede funcionar de manera normal (se pulsa y al soltar acaba el flanco de señal positivo) o con retención (estando a 0 se pulsa y se mantiene a 1 hasta que se vuelva a pulsar). En la figura 54 observamos un ejemplo configurado.

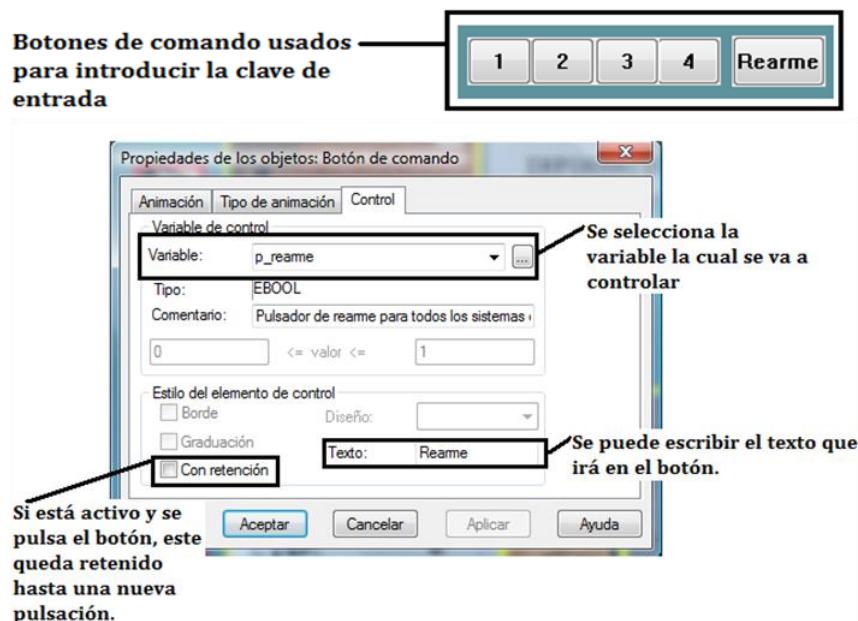


Figura 54: Ejemplo de configuración de botón de comando

Finalmente, en la figura 55 se pueden observar todos los elementos comentados con anterioridad.

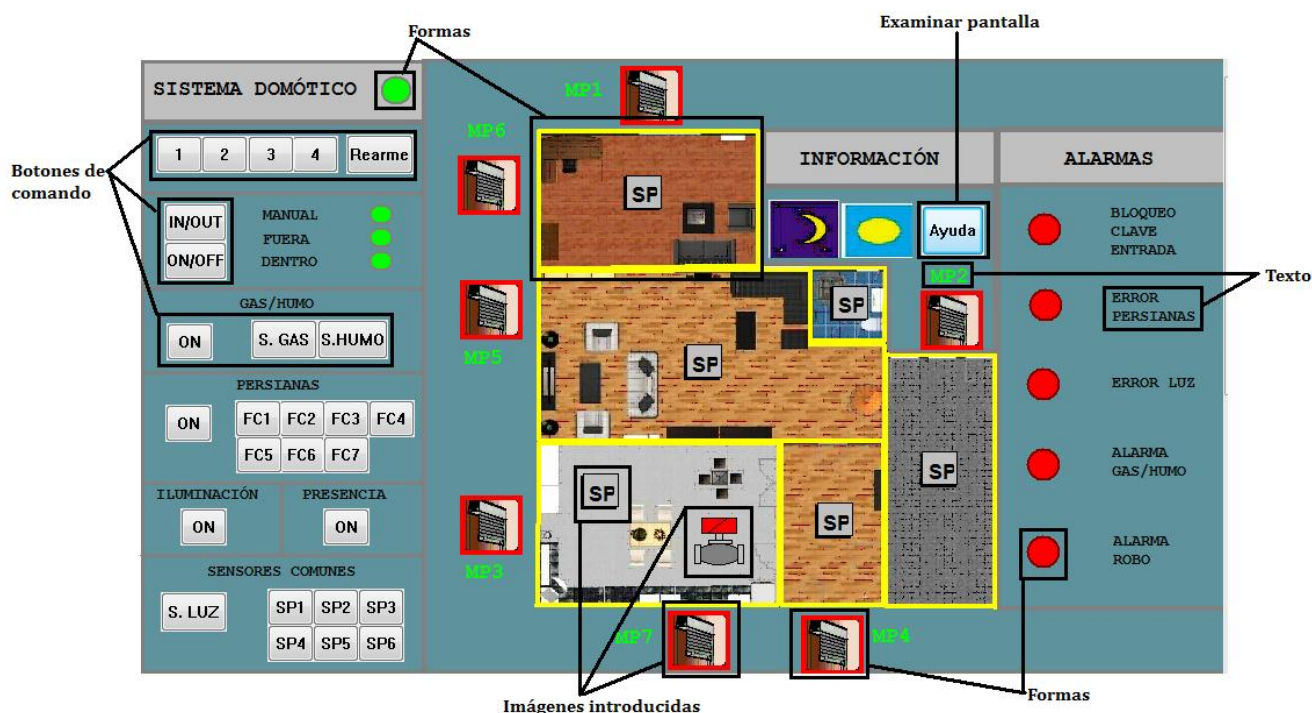


Figura 55: Elementos gráficos incorporados en el simulador domótico

Por último, destacar que Unity Pro cuenta con una biblioteca muy completa de símbolos e imágenes para facilitar el diseño de las pantallas de operador, entre sus archivos cuenta con todo tipo de accionadores, máquinas, autómatas, visualizadores, iconos, etc... Para poder visualizarlos e introducirlos en la pantalla debemos ir a Herramientas->librería de pantallas de operador, y automáticamente, se abrirá la pantalla que vemos en la figura 56.

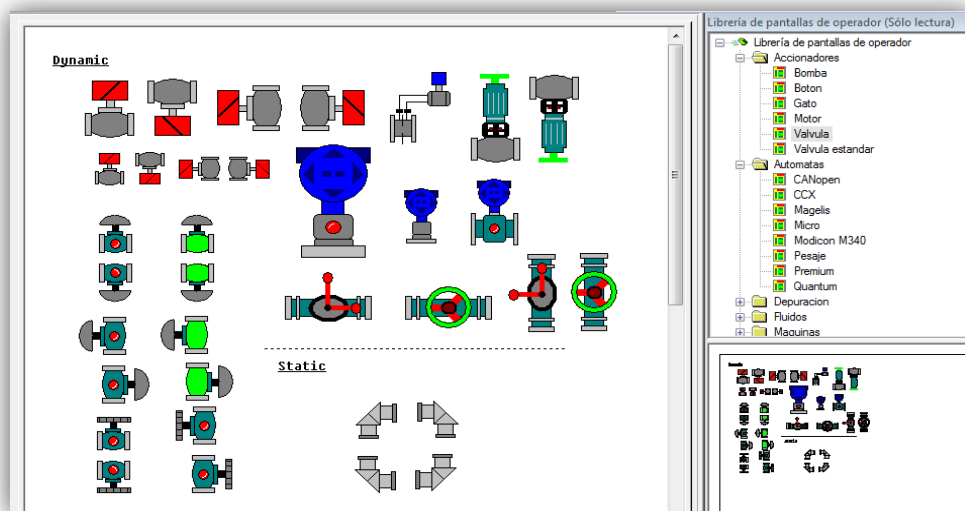


Figura 56: Librería de pantallas de operador Unity Pro XI



# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 5: SIMULACIÓN DEL DISEÑO***



El objetivo del simulador como se explicó en la introducción al presente trabajo es proveer al usuario de una herramienta gráfica sencilla, intuitiva y útil para representar procesos, de manera que los profesionales o alumnos que utilicen la aplicación puedan adquirir experiencia en manejo de máquinas, procesos, sistemas y ampliar sus conocimientos.

El simulador domótico ha sido creado con el objetivo de proveer al usuario de toda la información del estado de su vivienda (sensores, estado actuadores, errores de funcionamiento, etc...), mediante una pantalla de operador, de forma que, por ejemplo, desde un terminal o teléfono inteligente se puedan supervisar y controlar todos los sistemas de la casa, incluso estando a varios kilómetros de distancia. Actualmente es posible llevarlo a cabo gracias a los grandes avances realizados en la última década en comunicaciones e internet, los cuales, permiten la transmisión en tiempo real de gran cantidad de información. Este sistema se engloba principalmente en dos sectores, la industria y la docencia, ambos relacionados con la realización del trabajo.

En la actualidad, existen sistemas comerciales que realizan una función similar circulando por el mercado, aunque su grado de implantación en viviendas es reducido, debido principalmente, a su elevado coste, sin embargo, la necesidad de sistemas más sostenibles energéticamente hace de vital importancia la búsqueda y concepción de un software de bajo coste específico para domótica, buscando aumentar la implantación de este tipo de sistemas inteligentes, con el consecuente ahorro energético.

Tomando como base el presente trabajo y las condiciones que en él se han planteado, implantar dicho sistema en una vivienda real no acarrearía ningún tipo de problemas, de hecho, la programación ha sido diseñada para facilitar en todo lo posible la ampliación e incorporación de sistemas adicionales como, por ejemplo, control de puertas y ventanas, riego automático, puerta de garaje, climatización, comunicaciones, etc..., pudiendo incluir todo tipo de nuevos sistemas u optimizaciones de los ya existentes, e incluso, crear sistemas con un alto grado de personalización, cumpliendo así los deseos de los usuarios particulares.

En cuanto al ámbito docente, basándome en experiencias personales, este tipo de simuladores tiene un alto grado de aceptación por parte de los alumnos, debido a que ilustra conceptos teóricos explicados en el aula de una manera más amena e interactiva, pero no por ello menos eficaz. Este tipo de simuladores fuerzan el aprendizaje de conocimientos complementarios a los adquiridos en clases teóricas y aumentan la capacidad de razonamiento, planteando problemas complejos que el alumno tendrá que solucionar utilizando las herramientas aportadas por el simulador.



En el caso particular de la domótica, este simulador puede ser utilizado para explicar conceptos generales del funcionamiento de autómatas, relaciones entre entradas y salidas del sistema, tratamiento de errores, utilización de bloques de funciones (temporizadores, contadores...), pero sobretodo, para mostrar a los alumnos una visión simulada del funcionamiento de un sistema real, con el que posteriormente puedan encontrarse en su vida profesional.

Este capítulo se divide en tres métodos de simulación diferentes, mediante tablas de variables, pantalla de operador y pantalla de operador conectada con el autómata, en cada método realizaremos la simulación de una parte diferente del sistema domótico, intentando abarcar el funcionamiento de todos los sistemas incorporados en el simulador. El objetivo será observar la funcionalidad de los diferentes métodos de simulación disponibles, y posteriormente, poder analizar las ventajas e inconvenientes que presenta cada uno de ellos.

El hecho de no realizar la simulación completa con cada método, se debe a que la visualización y explicación de toda la simulación en los tres métodos empleados sería demasiado extensa y repetitiva, sin embargo, el perfecto funcionamiento del sistema si se comprobó en todos ellos.

Antes de entrar en detalle con cada método conviene explicar algunos conceptos comunes a la hora de realizar la simulación, entre ellos, se encuentra la utilización de la barra de herramientas específica para simulación, la cual, podemos observar en la siguiente figura:

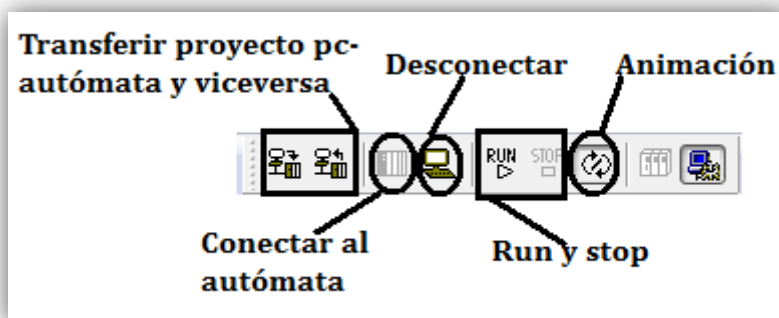


Figura 57: Barra de herramientas Api

Lo primero que debemos hacer es conectarnos al autómata, una vez conectados, si el proyecto contenido en el autómata es diferente al creado (porque se hayan realizado modificaciones o nunca se haya transferido), debemos pulsar el botón “transferir proyecto al autómata”, cuando el programa acabe de transferir, sólo nos queda darle a “Run”. Si queremos utilizar la pantalla de operador es muy importante activar los botones incorporados pulsando el botón que los habilita (ver figura 49).

## 5.1. Simulación utilizando tablas

En esta sección simularemos el funcionamiento desde inicio del sistema (incluyendo las alarmas que se pueden producir) hasta la selección y conmutación entre los diferentes modos de funcionamiento.

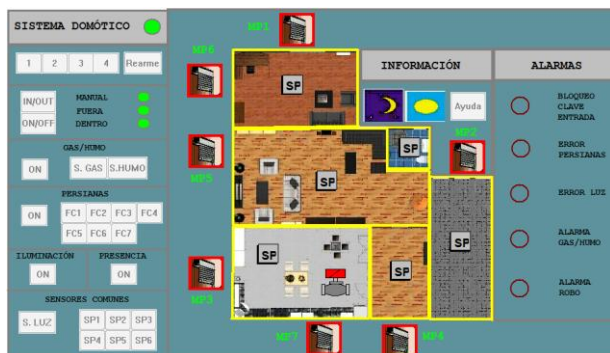


Figura 58: Sistema inactivo

Vemos el sistema inactivo, se pueden apreciar todos los elementos que incorpora, entre ellos, los que no son visibles durante la ejecución del programa como leds luminosos (verdes y rojos), avisos de luces encendidas (recuadros amarillos) y muchos más.

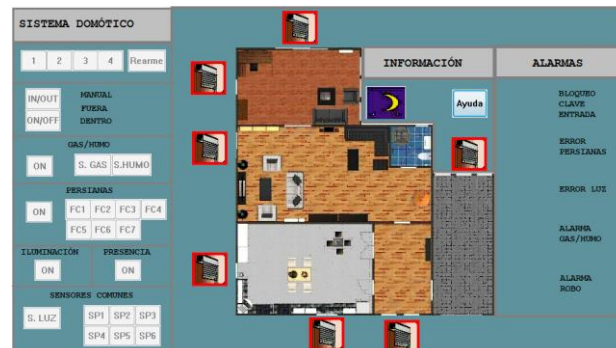


Figura 59: Sistema activo (sin botones)

En este instante, ponemos a funcionar el sistema domótico (pulsamos run) y observamos como muchos de los elementos visibles en la figura 58 han desaparecido.

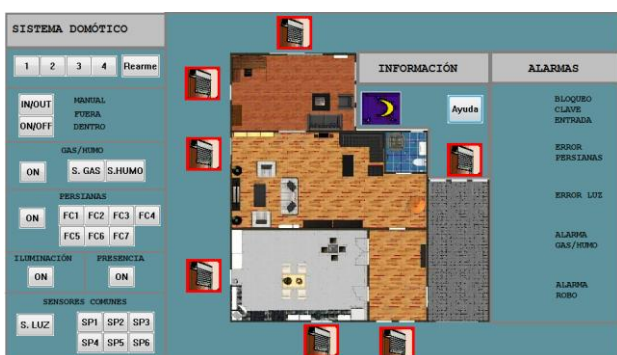


Figura 60: Sistema activo (con botones)

Para empezar con la simulación, primero debemos pulsar el botón correspondiente para activar los botones, vemos la diferencia entre la figura 60 (con botones activados) y 59 (sin ellos).

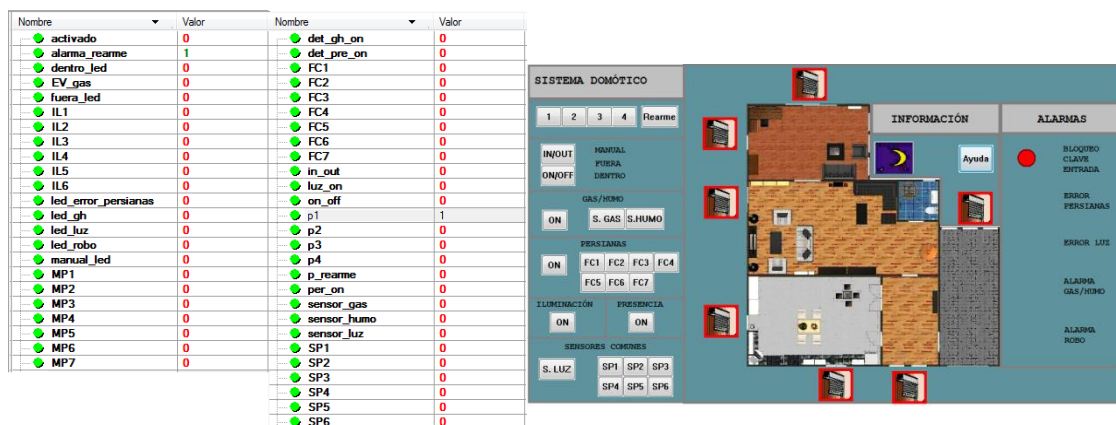


Figura 61: Alarma bloqueo clave de entrada

Para empezar a usar el sistema debemos introducir correctamente la clave numérica 4234, si la clave fuera errónea, sucedería lo que podemos ver en la figura 61, se activa el led de alarma “BLOQUEO CLAVE ENTRADA”, y en la tabla de salidas vemos activada la variable “alarma\_rearme”.

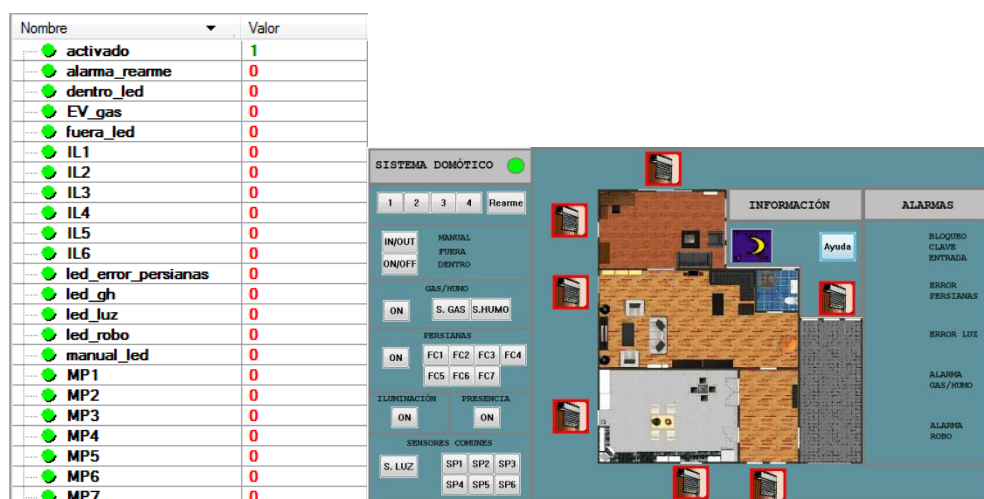


Figura 62: Sistema activado (con clave introducida)

Si la clave ha sido introducida de manera correcta, se observa junto al título “SISTEMA DOMÓTICO” un led verde que representa la correcta activación del sistema, además, en la tabla de salidas se puede ver activada la variable “activado”.



Figura 63: Modo manual activado

Pasamos entonces a elegir el modo de funcionamiento, elegiremos el modo “manual”, por ello, no pulsamos ni el botón “IN/OUT” ni “ON/OFF”, al entrar en el modo manual se encenderá un led verde justo al lado de la palabra “MANUAL” y se activará la variable “manual\_led” como vemos en la figura 63.



Figura 64: Modo fuera de casa activado

Ahora realizaremos una conmutación de modo pasando al modo fuera de casa, para ello, pulsamos el botón “IN/OUT” y esperamos unos segundos, automáticamente se apagará el led verde asociado al modo manual y se encenderá otro junto a la palabra “FUERA”, además, se activará la variable “fuera\_led”, la cual, representa la activación del modo fuera de casa.

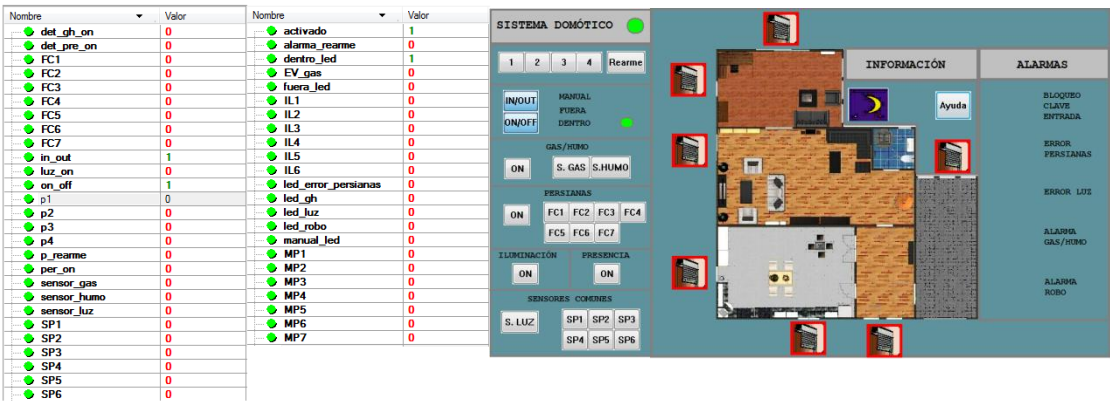


Figura 65: Modo dentro de casa activado

Por último, conmutaremos al modo dentro de casa, para ello, estando en el modo fuera de casa pulsamos “ON/OFF”, de manera que los dos botones se encuentren pulsados, al igual que en los demás casos se apagará el led del modo anterior y se encenderá un nuevo led verde junto a “DENTRO”, con la correspondiente activación de su variable de salida asociada “dentro\_led” (ver figura 65).



## 5.2. Simulación utilizando pantalla de operador

En este caso nos detendremos en el funcionamiento de algunos sistemas que incorpora el modo dentro de casa, por ejemplo, la alarma de gas/humo y la iluminación.



Figura 66: Modo dentro de casa activado



Figura 67: Gas/humo on

Como se ha visto en el apartado anterior con la simulación mediante tablas de animación, activamos el sistema introduciendo la clave de acceso, y, a continuación, la combinación de los botones “IN/OUT” y “ON/OFF” correspondiente al modo dentro de casa.

Una vez dentro del modo correspondiente, activamos el sistema de gas/humo pulsando el interruptor de encendido “ON”, de esta manera, si se produce una fuga será detectada a tiempo y el sistema podrá avisar al usuario.



Figura 68: Alarma gas/humo

Si se produce un flanco ascendente de los sensores de gas o humo (en este caso gas) se activa automáticamente la “ALARMA GAS/HUMO” y aparece en la cocina un icono con una válvula en rojo (cerrada), el cual, representa que se ha cerrado la electroválvula de gas por seguridad. Para volver al funcionamiento normal pulsamos “REARME”.

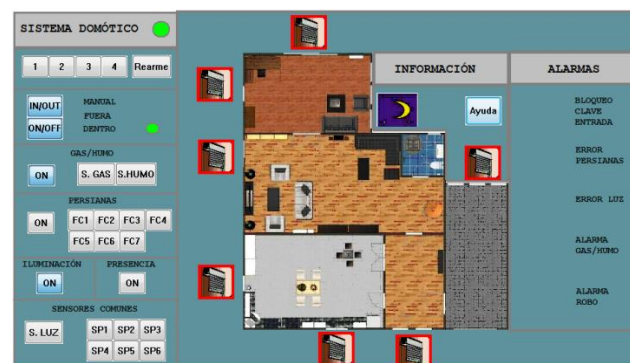


Figura 69: Iluminación on

Activamos la iluminación automática mediante el interruptor “ON” asociado a la iluminación, el cual, se encuentra situado en la botonera de igual manera que en el caso del sistema de detección de gas y humo.



Figura 70: Iluminación activada

Si es de noche (s.luz=0) vemos que aparece un icono de una luna, en este modo se encienden automáticamente todas las luces de la casa si se detecta movimiento en una habitación. Si observamos la imagen 70, vemos que se encuentran activas las luces del baño (SP1) y garaje (SP2), y que los sensores están detectando presencia (aparecen los iconos “SP” en cada habitación).



Figura 71: Apagado iluminación sin presencia

En este caso, vemos como se apaga la luz a los diez segundos si no se detecta presencia en las habitaciones, se puede observar como en el baño se ha dejado de detectar presencia (no aparece el icono “SP”) y si pasados los diez segundos no se detecta presencia, la luz se apagará.

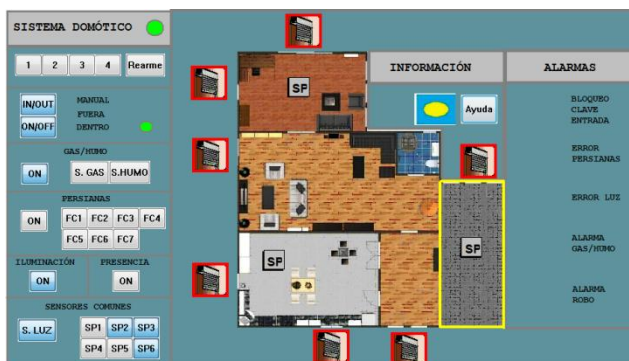


Figura 72: Iluminación selectiva

Si el sensor de luz detecta luminosidad suficiente, sólo funcionarán las luces de las habitaciones con poca luminosidad (baño y garaje). Si miramos la imagen 72, se observa como los sensores de presencia SP2, SP3 y SP4 detectan movimiento en la cocina, estudio y garaje, sin embargo, únicamente se enciende la luz del garaje (SP2).

### 5.3. Simulación conectando la pantalla de operador al autómeta

En este apartado se probará la pantalla de operador en el autómeta del laboratorio, la cual, se encuentra conectada al telefast (tarjeta con 16 pulsadores y leds colocada a cierta distancia del pc, ver figura 73) para poder interactuar con el autómeta. El principal problema que ha surgido en esta simulación ha sido la cantidad de entradas y salidas necesarias para simular el sistema, éstas se encuentran limitadas a 16 por los módulos de entradas y salidas existentes en el autómeta, sin embargo, el simulador utiliza 27 variables de entrada, como solución, se optó por dividir las variables en 3 módulos (la mínima cantidad posible para su funcionamiento) y reconfigurar tres veces el mismo módulo con las diferentes configuraciones. En las siguientes tablas observaremos como se han dividido las variables.

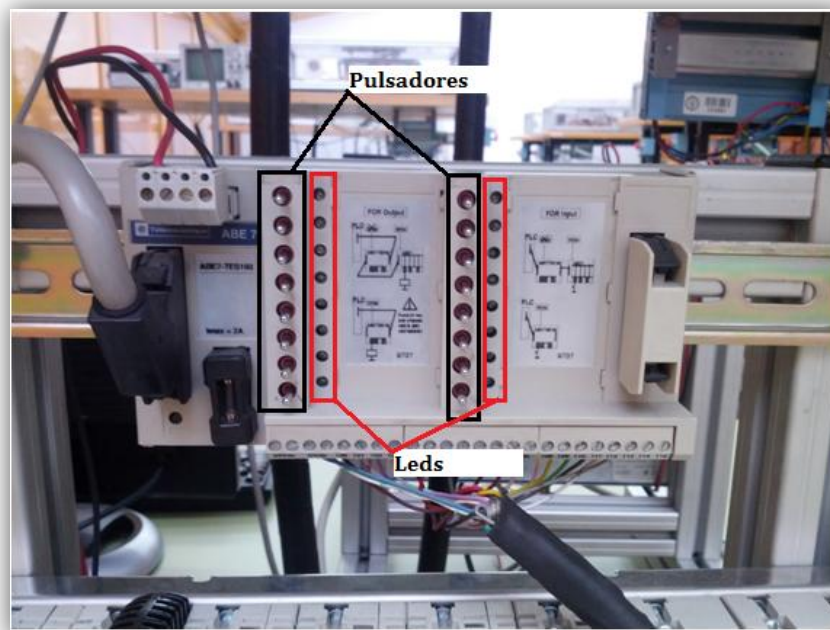


Figura 73: Telefast



MÓDULO 1				MÓDULO 2			
DEY 1		DSY 1		DEY 2		DSY 2	
P1	%I0.1.0	LED_ACTIVADO	%Q0.2.0	P1	%I0.3.0	LED_ACTIVADO	%Q0.4.0
P2	%I0.1.1	LED_MANUAL	%Q0.2.1	P2	%I0.3.1	LED_MANUAL	%Q0.4.1
P3	%I0.1.2	LED_FUERA	%Q0.2.2	P3	%I0.3.2	LED_FUERA	%Q0.4.2
P4	%I0.1.3	LED_DENTRO	%Q0.2.3	P4	%I0.3.3	LED_DENTRO	%Q0.4.3
P_REARME	%I0.1.4	MP1	%Q0.2.4	P_REARME	%I0.3.4	IL1	%Q0.4.4
IN/OUT	%I0.1.5	MP2	%Q0.2.5	IN/OUT	%I0.3.5	IL2	%Q0.4.5
ON/OFF	%I0.1.6	MP3	%Q0.2.6	ON/OFF	%I0.3.6	IL3	%Q0.4.6
FC1	%I0.1.7	MP4	%Q0.2.7	SP1	%I0.3.7	IL4	%Q0.4.7
FC2	%I0.1.8	MP5	%Q0.2.8	SP2	%I0.3.8	IL5	%Q0.4.8
FC3	%I0.1.9	MP6	%Q0.2.9	SP3	%I0.3.9	IL6	%Q0.4.9
FC4	%I0.1.10	MP7	%Q0.2.10	SP4	%I0.3.10	ALARMA_LUZ	%Q0.4.10
FC5	%I0.1.11	ALARMA_GEN	%Q0.2.11	SP5	%I0.3.11	ALARMA_GEN	%Q0.4.11
FC6	%I0.1.12	ALARMA_PER	%Q0.2.12	SP6	%I0.3.12	ALARMA_ROBO	%Q0.4.12
FC7	%I0.1.13			ON_LUZ	%I0.3.13		
ON_PER	%I0.1.14			ON_PRE	%I0.3.14		
S.LUZ	%I0.1.15			S.LUZ	%I0.3.15		

Tabla 3: Configuración 1 y 2 módulo hardware

MÓDULO 3			
DEY 3		DSY 3	
P1	%I0.5.0	LED_ACTIVADO	%Q0.6.0
P2	%I0.5.1	LED_MANUAL	%Q0.6.1
P3	%I0.5.2	LED_FUERA	%Q0.6.2
P4	%I0.5.3	LED_DENTRO	%Q0.6.3
P_REARME	%I0.5.4	ALARMA_GEN	%Q0.6.4
IN/OUT	%I0.5.5	ALARMA_G/H	%Q0.6.5
ON/OFF	%I0.5.6	EV GAS	%Q0.6.6
ON G/H	%I0.5.7		
S.GAS	%I0.5.8		
S.HUMO	%I0.5.9		

Tabla 4: Configuración 3 módulo hardware

Otra manera posible de realizar lo explicado en el párrafo anterior, hubiera sido la utilización de dos módulos de entradas y dos de salidas, con los cuales, dispondríamos de las entradas y salidas suficientes para el funcionamiento de nuestro sistema. Esto no se llevó a cabo, debido a los equipos son usados por estudiantes de la universidad en prácticas obligatorias y era inviable realizar dicha configuración al autómata, por lo tanto, la única solución era configurar el único módulo varias veces.



Figura 74: Modo manual fuera de casa activado



Figura 75: Bajada persianas y luces

Como se vio en la simulación mediante tablas de animación, activamos el sistema introduciendo la clave de acceso, y, a continuación, la combinación de los botones “IN/OUT” y “ON/OFF” correspondiente al modo fuera de casa.

Dentro del modo fuera de casa se bajan y apagan luces y persianas de manera automática. Si observamos la imagen 75, vemos que una de las persianas sigue subida (tiene un recuadro verde), de manera que se ha activado la señal MP<sub>1</sub>, la cual, simboliza que el motor de la persiana está en marcha (cambiando el estado de la persiana).



Figura 76: Alarma error bajada de persianas



Figura 77: Alarma robo

Si por alguna razón, pasado 1 minuto no se consigue bajar la persiana, saltará la alarma “ERROR PERSIANAS” para avisar de una anomalía del sistema. Para salir y volver a un correcto funcionamiento tendremos que pulsar el botón de “REARME”.

Al igual que otros sistemas como la alarma de gas/humo, para activar la alarma ante intrusos debemos pulsar el interruptor de encendido “ON” situado en la botonera del simulador. Una vez hecho esto, si nos fijamos en la imagen 77, observamos que si se detecta presencia en una habitación saltará la “ALARMA ROBO”.

## 5.4. Comparación entre simulaciones

Observando las explicaciones y los resultados obtenidos en los apartados anteriores, nos damos cuenta de las ventajas de los simuladores (pantalla de operador) a la hora de realizar la misma función que las tablas de variables, e incluso, que el mismo autómatas.

En cuanto a funcionalidad, los tres métodos son muy semejantes, ya que todos ellos permiten utilizar el sistema sin problemas, sin embargo, es cierto, que cada uno tiene particularidades que les hacen más atractivos para determinados usos, por ejemplo, la pantalla de operador ofrece un botón de conexión entre varias pantallas, de manera que se podrían realizar una cantidad determinada de pantallas, digamos 5, e ir navegando indistintamente por ellas durante la ejecución del programa, pudiendo controlar mucha cantidad de procesos diferentes a la vez. Además de esto, el factor de representación gráfica mejora muchísimo la comprensión de la información y la sencillez de uso, siempre teniendo en cuenta que el diseño de la pantalla es el adecuado.

Las tablas de variables tienen la ventaja de poder forzar o modificar los valores de todas las variables incluidas en ellas, incluso variables que en el simulador no podemos controlar, aquí si son modificables, por ello, su uso está más enfocado a procesos de prueba y error realizados durante la programación de etapas y transiciones, debido también, a que la observación del cambio de estado de las variables es menos intuitivo que en las pantallas de operador, y más aún, cuando el número de variables es elevado (por ejemplo, no es lo mismo ver un led parpadeante de error, que tener que buscar en un listado de 40 variables un número iluminado).

Por último, cuando conectamos la pantalla de operador al autómatas, nos damos cuenta que el telefast aunque realiza la misma función que la botonera y las formas gráficas incluidas en la pantalla de operador, sigue siendo menos intuitivo. Una de las razones es el hecho de no contener el nombre de la variable asignada ni en pulsadores, ni en leds, ésto, unido a que se encuentra a la distancia suficiente del ordenador para resultar incómodo, dificulta la simulación y hace que sea menos intuitiva que únicamente utilizando el simulador.

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 6: CONCLUSIÓN***



## 6.1. Conclusiones generales

Sin duda, se ha demostrado con creces que el software Unity Pro es una potente herramienta de programación y diseño de procesos, además, cuenta con facilidad de aprendizaje y uso, permitiendo ahorrar tiempo y dinero a los ingenieros a la hora de realizar múltiples tareas como ejemplificaciones de sistemas reales de control, simplificaciones de sistemas complejos, optimizaciones de sistemas reales, etc...

El objetivo inicial planteado consistía en realizar un simulador que proporcionara una solución a un problema de automatización, en este caso, se eligió la automatización de una vivienda. Comprobando los resultados obtenidos en la realización del proyecto se concluye que el diseño del simulador es el correcto, su funcionalidad es acorde con su ámbito de utilización y que cumple todas las condiciones iniciales que fueron planteadas para un sistema de estas características. En la realización del simulador se procuró agrupar todos los botones en una misma zona de la pantalla reservada para ello, facilitando su funcionalidad pero sin influir en exceso en su estética. Para facilitar su uso, los botones fueron divididos en apartados dependiendo a los sistemas que pertenecieran.

En cuanto a la parte gráfica del simulador, se utilizaron diferentes elementos con el objetivo de poder interactuar con él, y que éste, activara las salidas correspondientes a las entradas introducidas. Se procuró contar con un alto grado de representación de la realidad y se consiguió que la interpretación del sistema fuera intuitiva y de un uso realmente sencillo.

A diferencia del software PL7 Junior utilizado anteriormente en las aulas de automatización, Unity Pro abre una nueva posibilidad de trabajo mediante simuladores gráficos, esto permite simular los proyectos simplemente usando un ordenador, sin necesidad de utilizar un autómata, a su vez, permite trabajar desde casa a los alumnos de la universidad, comprobando problemas realizados en el aula, o incluso, sirviendo de apoyo en el desarrollo de las prácticas de la asignatura, también, como se comentó con anterioridad, se ha ganado mucho en sencillez de manejo frente a los software anteriores.

## 6.2. Posibles mejoras del simulador

La realización del simulador se ha visto limitada por el número de entradas y salidas disponibles en el autómata, inicialmente se pensó en realizar un sistema domótico con más sistemas incorporados, intentando tocar las cuatro áreas funcionales de la domótica, confort, seguridad, comunicaciones y gestión energética. Finalmente, teniendo en cuenta la citada limitación de entradas y salidas, se pensó en priorizar e introducir los elementos más imprescindibles o más comunes en este tipo de sistemas, todo ello, intentando siempre incorporar el máximo número de áreas funcionales.

El sistema diseñado se centra principalmente en seguridad y confort de la vivienda, incluyendo también algún sistema con funcionalidades relacionadas con una buena gestión energética. Se ha realizado una programación fácilmente ampliable pensando en una posterior ampliación e incorporación de mayor número de funcionalidades, de manera que con poco tiempo de trabajo se puedan incorporar fácilmente más funciones.

En cuanto al diseño gráfico, el simulador ya se encuentra lo suficientemente recargado de botones, alarmas, iconos, etc... , por esta razón, en mi opinión, la solución sería retocar la pantalla principal para ver únicamente el estado genérico de todos los subsistemas (Encendido, apagado, funcionando...), y aparte de esa pantalla principal, crearía una pantalla de operador específica para cada subsistema, de manera que se puedan controlar todos los sistemas y desplazarnos entre las diferentes pantallas de operador durante la ejecución del programa. También, cabe añadir que este tipo de configuración sería viable si la cantidad de sistemas incorporados es alta, debido a la imposibilidad de representar toda la información necesaria en una sola pantalla.

Por último, esclarecer que la forma de realizar el diseño de las diferentes alarmas, avisos, iconos y del simulador en general, ha sido pensado acorde con la funcionalidad requerida, intentando optimizar al máximo la facilidad de su uso y comprensión para el usuario, sin embargo, esto no quiere decir que no se puedan cambiar ciertos elementos y conseguir una solución más eficiente. De igual manera, la programación realizada en cada subsistema puede ser cambiada, ampliada y optimizada a gusto del usuario final del sistema, ya que lo explicado en este proyecto simplemente es una solución de las múltiples que existen.

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 7: ANEXOS***



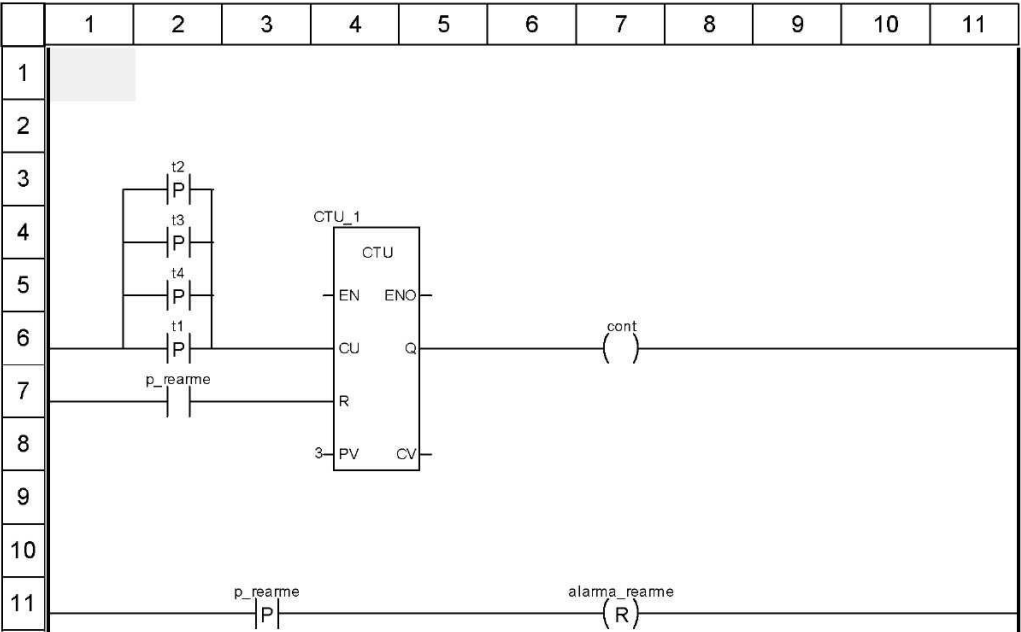
## 7.1. Anexo I: programación de etapas y transiciones en lenguaje Ladder (LD)

En el presente anexo se añade toda la información complementaria a la explicación realizada en el capítulo 4 sobre la programación del simulador. Se ha realizado una tabla con la relación existente entre las transiciones y etapas nombradas en la explicación y el nombre correspondiente de las acciones programadas en el software, de manera que sea más sencillo localizar una parte específica de la programación. Además de esto y con el fin de facilitar más aún el entendimiento, se ha colocado junto a cada relación un hipervínculo que redirige a la página correspondiente.

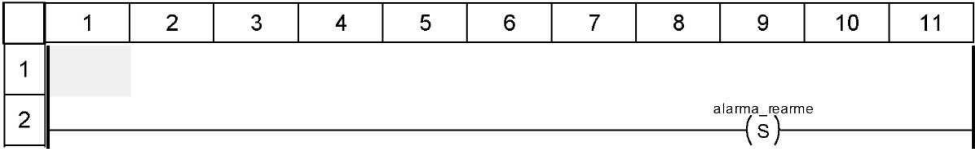


SFC PRINCIPAL		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Decidir_modo	Elegir	<a href="#">1</a>
Fuera_casa	Fuera_casa	<a href="#">2</a>
manual	manual	<a href="#">3</a>
Dentro_casa	Dentro_casa	<a href="#">4</a>
Espera1	Etapa_manual	<a href="#">5</a>
Espera2	Salir_etapa	<a href="#">6</a>
Espera3	Salir_etapa_dentro	<a href="#">7</a>
Cambiar	Cambio_fuera	<a href="#">8</a>
Seguir_cambiar	Salir_dentro_1	<a href="#">9</a>
CLAVE_INICIO		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Inicio	C	<a href="#">10</a>
Alarma_amarilla	Alarma	<a href="#">11</a>
t1	t1	<a href="#">12</a>
t2	t2	<a href="#">13</a>
t3	t3	<a href="#">14</a>
t4	t4	<a href="#">15</a>
APAGAR_LUCES		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Comprobar_luces_apagadas_ext	Apagar_luces_ext	<a href="#">16</a>
Alarma_luces_ext	Led_luces_ext	<a href="#">17</a>
BAJAR_PERSIANAS_EXT		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
bajar_todas_persianas_ext	Comprobar_bajada	<a href="#">18</a>
Problema_persianas	Led_persianas_ext	<a href="#">19</a>
DET_PRESENCIA		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Sensores_presencia	Sensor_presencia	<a href="#">20</a>
Alarma_presencia	Alarma_presencia_seccion	<a href="#">21</a>
DET_GAS_HUMO_EXT y DET_GAS_HUMO		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Sensor_gas_ext o Sensor_gas_humo	Detectar_gas_o_humo	<a href="#">22</a>
Alarma_gas_ext o Alarma_gas	Alarma_gas_humo	<a href="#">23</a>
UP_DOWN_PERSIANAS		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Bajada	Comprobar_bajada_dentro	<a href="#">24</a>
Alarma_per_bajada	Led_persianas_ext	<a href="#">25</a>
Subida	comprobar_subida	<a href="#">26</a>
Alarma_per_subida	Led_persianas_ext	<a href="#">27</a>
ILUMINACION		
ETAPA/TRANSICIÓN	ACCIÓN	HIPERVÍNCULO
Todo_con_luz	Encender_luz_total	<a href="#">28</a>
Bano_garaje	Encender_luz_parcial	<a href="#">29</a>

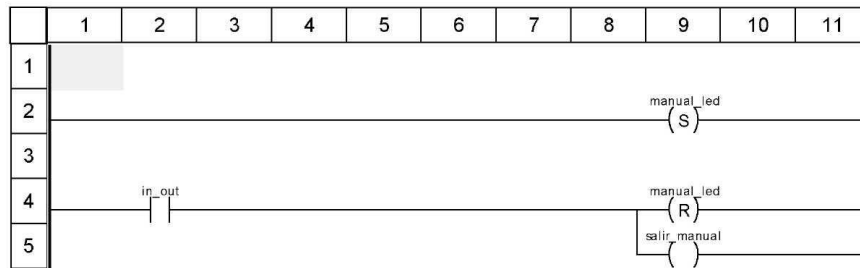
c <Acción> : [MAST - control\_de\_acceso



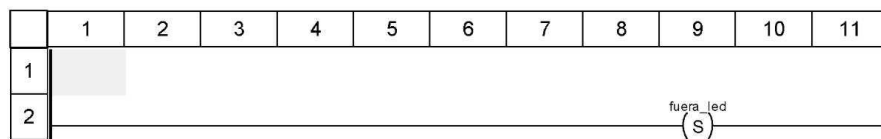
alarma <Acción> : [MAST - control\_de\_acceso]



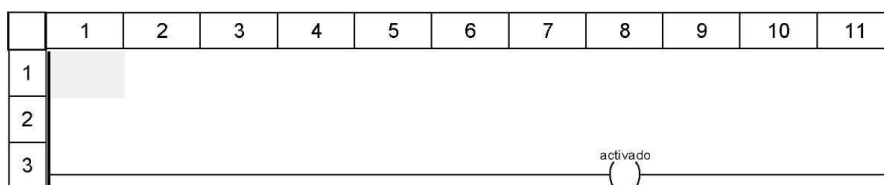
## etapa\_manual <Acción> : [MAST - control\_de\_acceso]



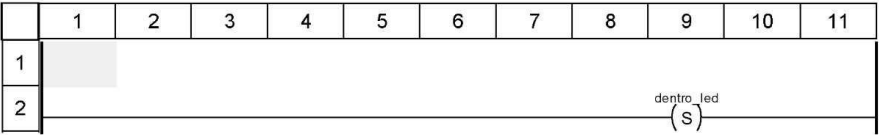
## salir\_etapa <Acción> : [MAST - control\_de\_acceso]



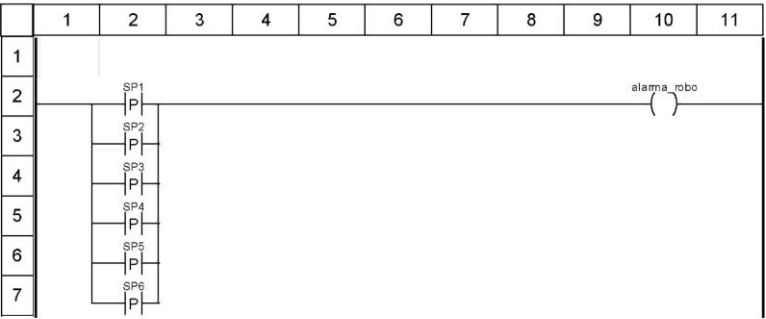
## elegir <Acción> : [MAST - control\_de\_acceso]



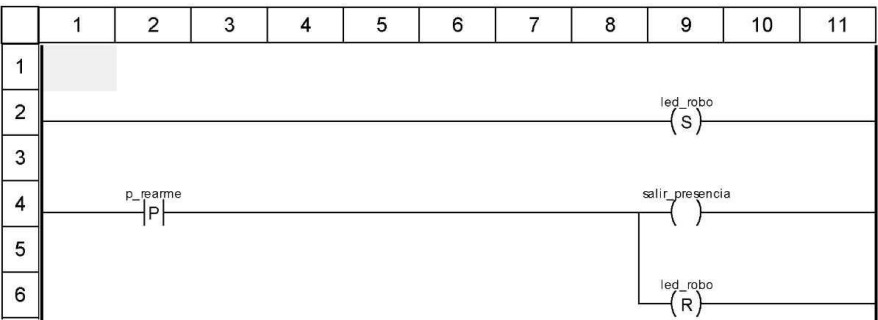
**salir\_etapa\_dentro <Acción> : [MAST - control\_de\_acceso]**



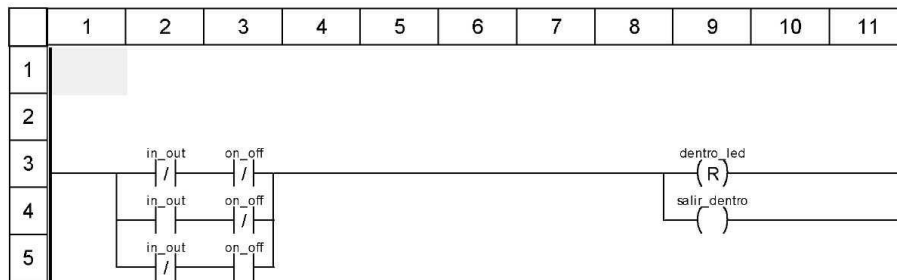
**sensor\_presencia <Acción> : [MAST - control\_de\_acceso]**



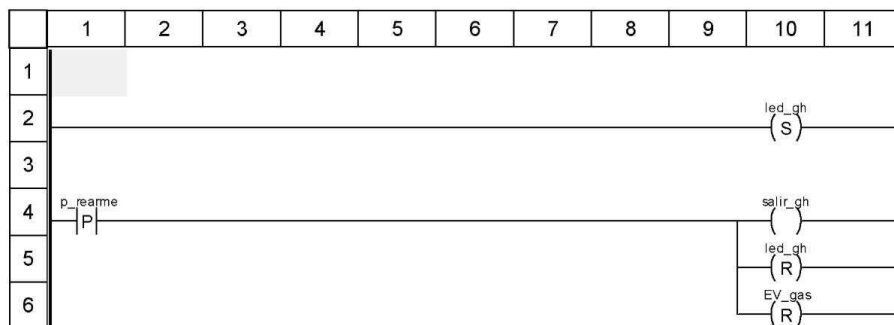
**alarma\_presencia\_seccion <Acción> : [MAST - control\_de\_acceso]**



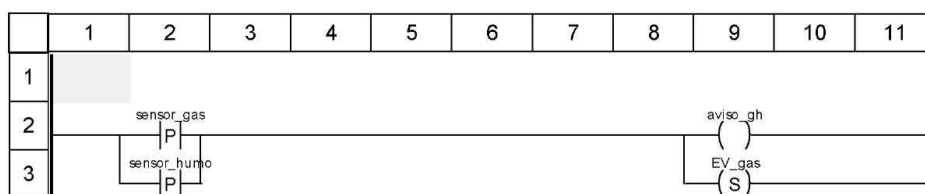
## salir\_dentro\_1 <Acción> : [MAST - control\_de\_acceso]



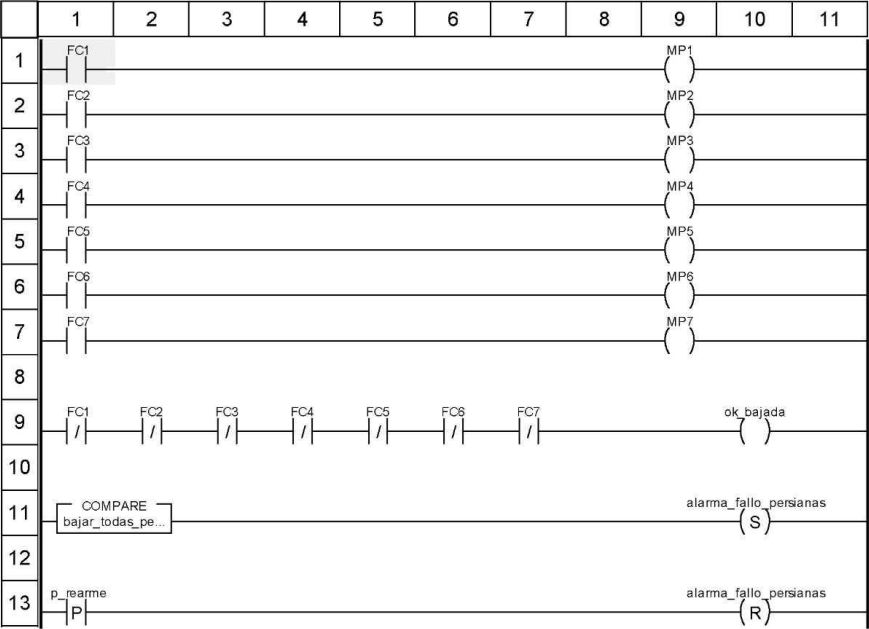
## alarma\_gas\_humo <Acción> : [MAST - control\_de\_acceso]



## detectar\_gas\_o\_humo <Acción> : [MAST - control\_de\_acceso]



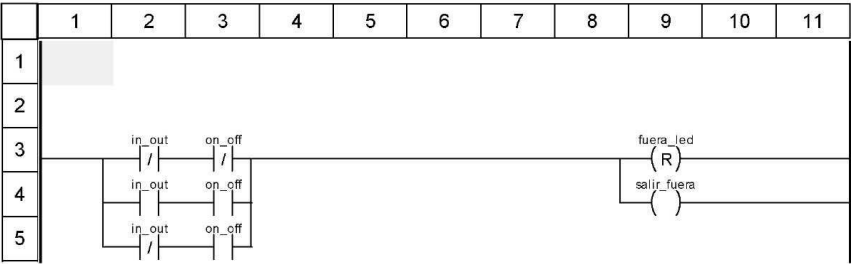
comprobar\_bajada <Acción> : [MAST - control\_de\_acceso]



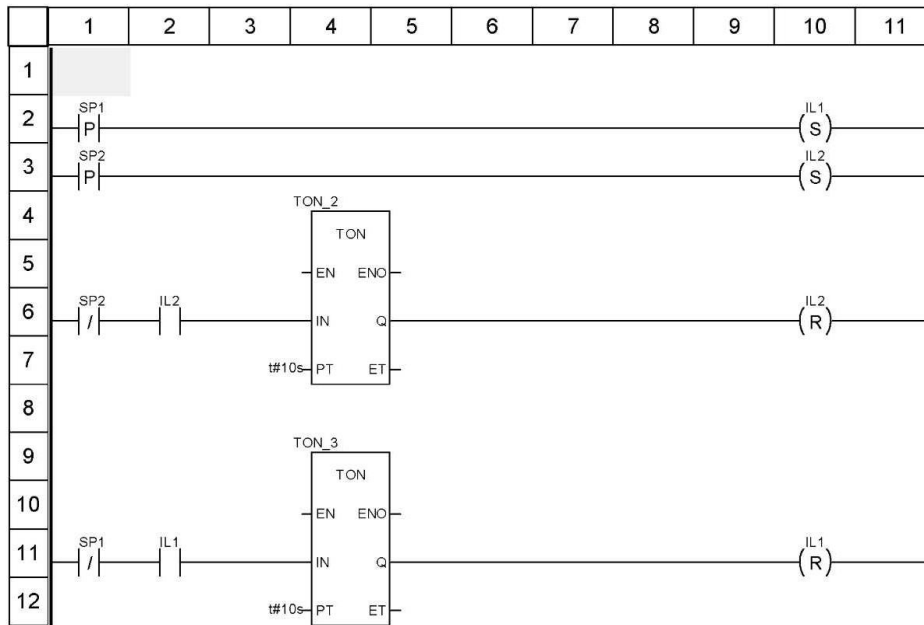
Etiquetas truncadas:

Etiqueta	Posición(es)
bajar_todas_persianas_ext.tmaxErr=1	(1, 11)

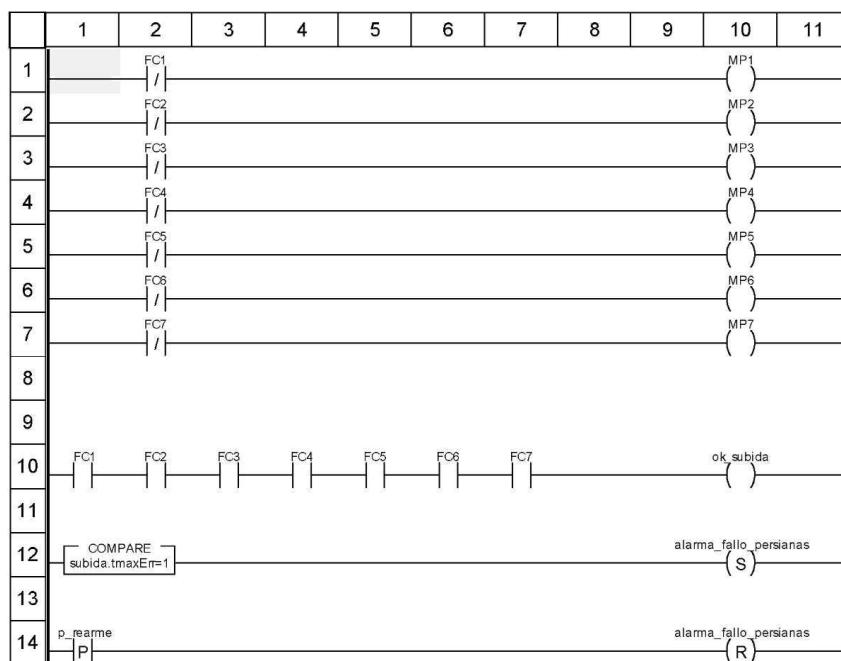
cambio\_fuera <Acción> : [MAST - control\_de\_acceso]



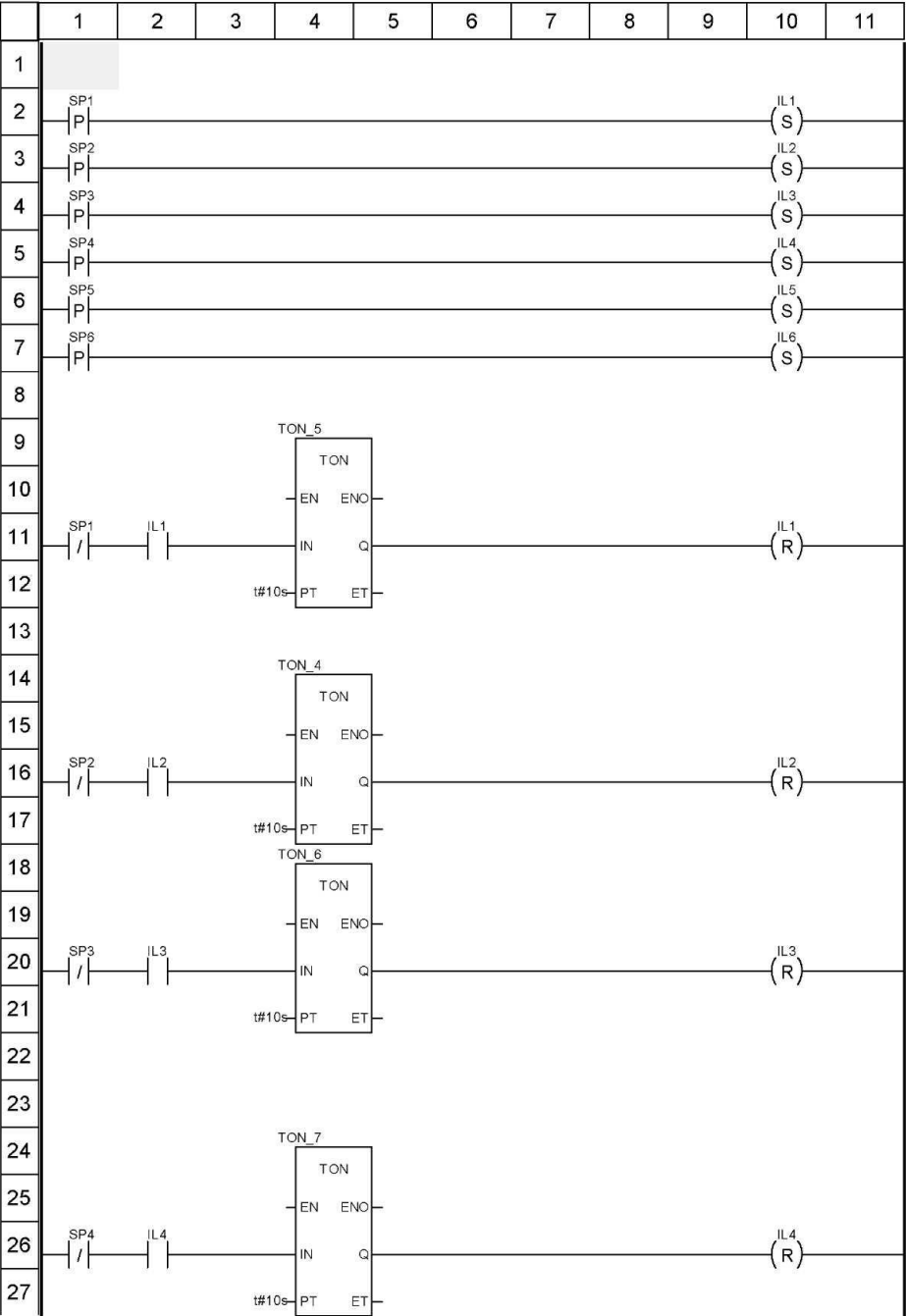
## encender\_luz\_parcial <Acción> : [MAST - control\_de\_acceso]



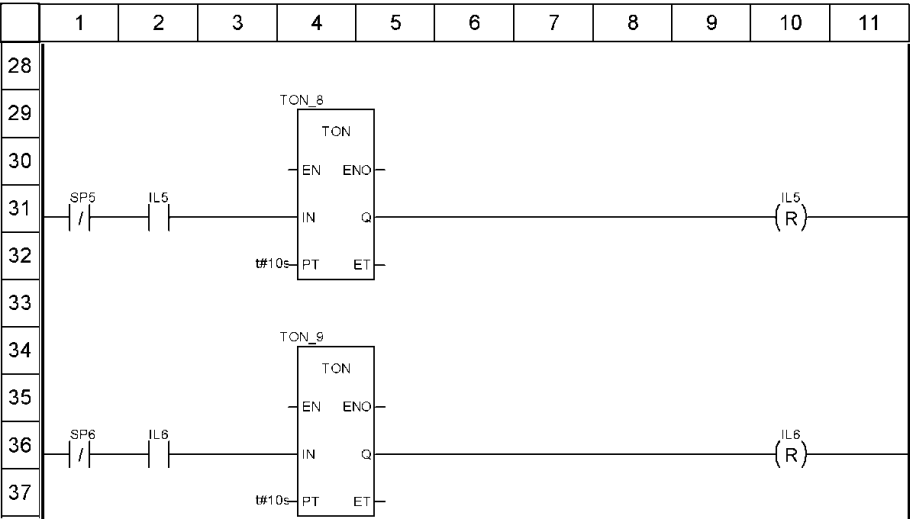
## comprobar\_subida <Acción> : [MAST - control\_de\_acceso]



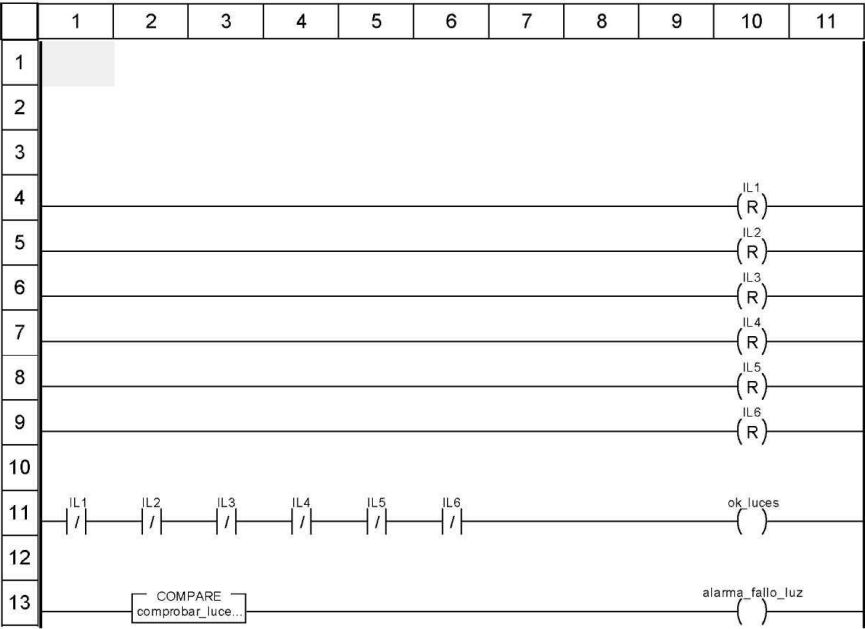
**encender\_luz\_total <Acción> : [MAST - control\_de\_acceso]**







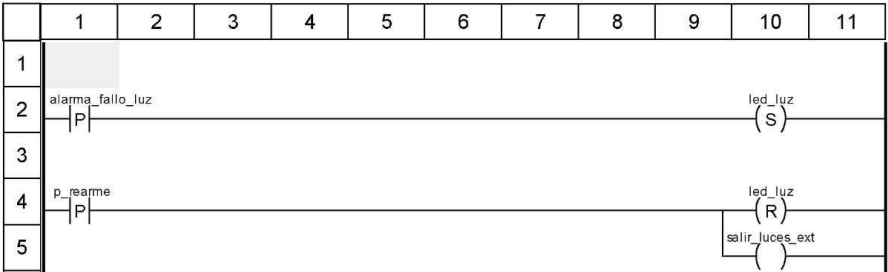
**apagar\_luces\_ext <Acción> : [MAST - control\_de\_acceso]**



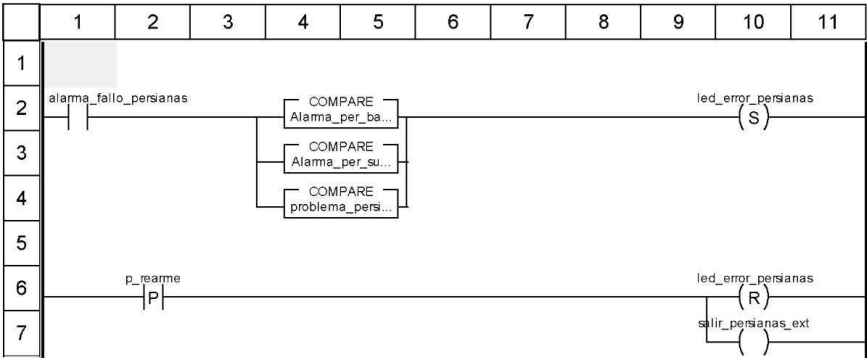
**Etiquetas truncadas:**

Etiqueta	Posición(es)
comprobar_luces_apagadas_ext.tmaxEn=1	(2, 13)

led\_luces\_ext <Acción> : [MAST - control\_de\_acceso]



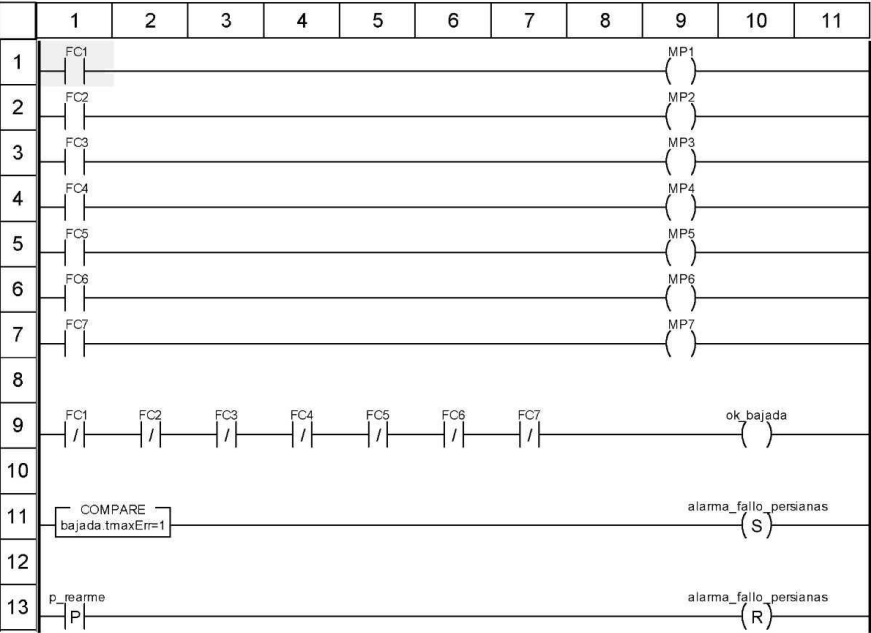
led\_persianas\_ext <Acción> : [MAST - control\_de\_acceso]



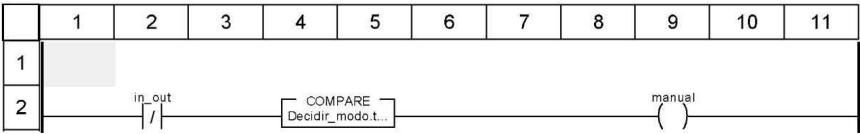
Etiquetas truncadas:

Etiqueta	Posición(es)
Alarma_per_bajada.t>=t#1s	(4, 2)
Alarma_per_subida.t>=t#1s	(4, 3)
problema_persianas.t>=t#1s	(4, 4)

**comprobar\_bajada\_dentro <Acción> : [MAST - control\_de\_acceso]**



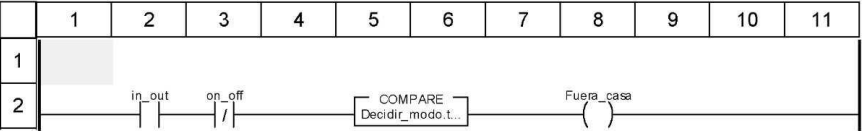
**manual <Transición> : [MAST - control\_de\_acceso]**



Etiquetas truncadas:

Etiqueta	Posición(es)
Decidir_modos.tmaxEn=1	(4, 2)

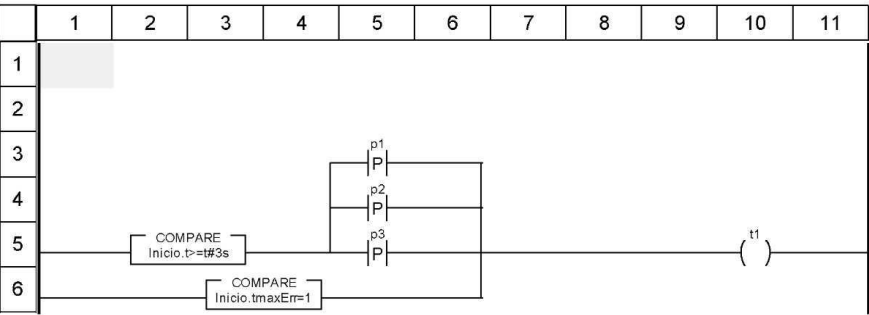
Fuera\_casa <Transición> : [MAST - control\_de\_acceso]



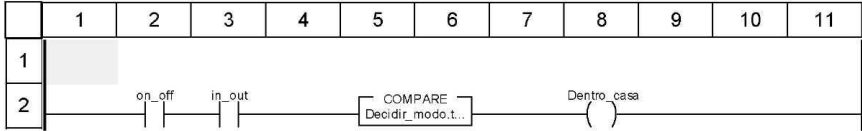
Etiquetas truncadas:

Etiqueta	Posición(es)
Decidir_modos.tmaxErr=1	(5, 2)

t1 <Transición> : [MAST - control\_de\_acceso]



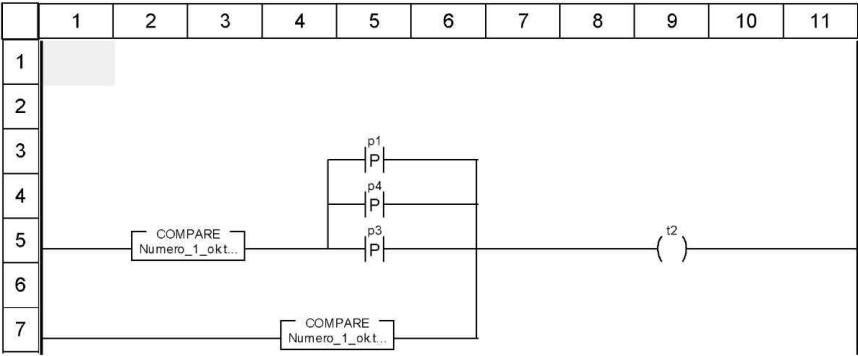
Dentro\_casa <Transición> : [MAST - control\_de\_acceso]



Etiquetas truncadas:

Etiqueta	Posición(es)
Decidir_modos.tmaxErr=1	(5, 2)

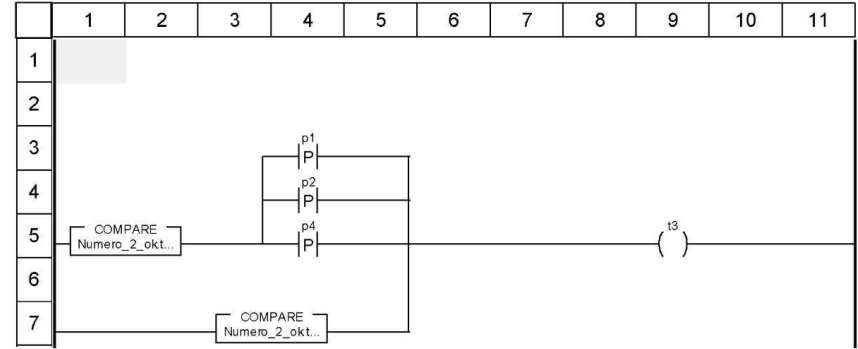
t2 <Transición> : [MAST - control\_de\_acceso]



Etiquetas truncadas:

Etiqueta	Posición(es)
Numero_1_ok.t>=t#3s	(2, 5)
Numero_1_ok.tmaxErr=1	(4, 7)

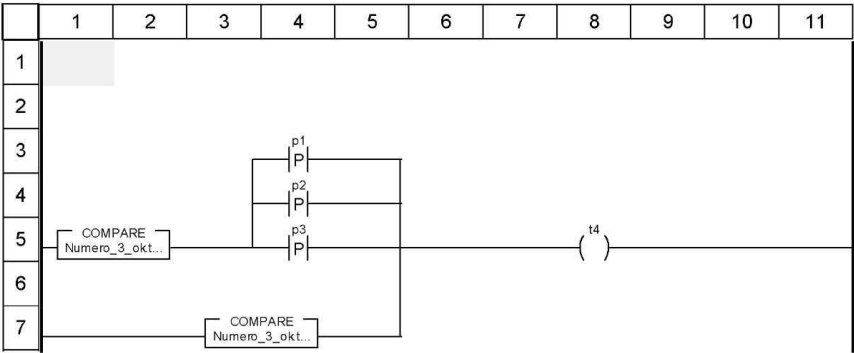
t3 <Transición> : [MAST - control\_de\_acceso]



Etiquetas truncadas:

Etiqueta	Posición(es)
Numero_2_ok.t>=t#3s	(1, 5)
Numero_2_ok.tmaxErr=1	(3, 7)

t4 <Transición> : [MAST - control\_de\_acceso]



Etiquetas truncadas:

Etiqueta	Posición(es)
Numero_3_ok.t>=t#3s	(1, 5)
Numero_3_ok.tmaxErr=1	(3, 7)

# **SIMULACIÓN DE LA AUTOMATIZACIÓN DE PROCESOS EN UNITY PRO: VIVIENDA DOMÓTICA**

***CAPITULO 8: BIBLIOGRAFÍA***



- [1] Real Academia Española de la Lengua. *Diccionario de la Real Academia Española*. 22ª ed., Espasa\_Calpe, Barcelona, 2001.
- [2] R. P. Moreno, *Ingeniería de la Automatización Industrial*, 2ª ed., Ra-Ma, 2004.
- [3] Wikipedia, *La enciclopedia libre* [en línea]  
<[http://es.wikipedia.org/wiki/Automatizaci%C3%B3n\\_industrial](http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial)> [Consulta: 3 de Junio de 2013]
- [4] L. Molina, *Instalaciones Automatizadas en Viviendas y Edificios*, Mc Graw Hill, 2005.
- [5] Asociación Española de Domótica. (2011). *Tendencias del mercado español de domótica e inmótica* [en línea]. Disponible en web:  
[http://www.cedom.es/fitxers/documents/publicacions/CEDOM\\_Estudio%20Tendencias%20Mercado%202011.pdf](http://www.cedom.es/fitxers/documents/publicacions/CEDOM_Estudio%20Tendencias%20Mercado%202011.pdf) [Consulta: 3 de Junio de 2013]
- [6] Wikipedia, *La enciclopedia libre* [en línea]  
<[http://es.wikipedia.org/wiki/Automatizaci%C3%B3n\\_industrial](http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial)> [Consulta: 3 de Junio de 2013]
- [7] W. Glass-Husain. *What is a simulation?. The drive to learn, foro business simulations*. [Consulta: 3 de Junio de 2013]
- [8] Laboratorio de Simuladores en Administración y Gerencia. *Historia y eficacia de la simulación* [en línea]. Disponible en web: < <http://www.gerentevirtual.com/historia.asp>> [Consulta: 3 de Junio de 2013]
- [9] Wikipedia, *La enciclopedia libre* [en línea]  
<[http://es.wikipedia.org/wiki/Automatizaci%C3%B3n\\_industrial](http://es.wikipedia.org/wiki/Automatizaci%C3%B3n_industrial)> [Consulta: 3 de Junio de 2013]
- [10] *Schneider Electric* [en línea]. <<http://www.schneiderelectric.com>> [Consulta: 3 de Junio de 2013]
- [11] Schneider Electric. (2010). *Lenguajes y estructura del programa. Manual de referencia* [en línea]. Disponible en web:  
<[http://www.schneiderelectric.com/resources/sites/SCHNEIDER\\_ELECTRIC/content/live/FAQS/30000/FA30629/es\\_ES/Unity%20v50%20%20Lenguajes%20y%20estructura%20del%20programa.pdf](http://www.schneiderelectric.com/resources/sites/SCHNEIDER_ELECTRIC/content/live/FAQS/30000/FA30629/es_ES/Unity%20v50%20%20Lenguajes%20y%20estructura%20del%20programa.pdf)> [Consulta: 3 de Junio de 2013]
- [12] I.Bravo. *Entornos de automatizaciones industriales* [en línea] [Universidad de Alcalá, España]. Disponible en web:  
<[http://193.146.57.132/depeca/repositorio/ asignaturas/201608/T3\\_VIJE0\\_2\\_comunic\\_PLC.pdf](http://193.146.57.132/depeca/repositorio/ asignaturas/201608/T3_VIJE0_2_comunic_PLC.pdf)> [Consulta: 3 de Junio de 2013]
- [13] Instituto Schneider Electric de Formación. (2008). *Manual de formación Unity Pro* [en línea]. Disponible en web: < [http://www.adtecnologias.com/Catalogos/Manual\\_Unity.pdf](http://www.adtecnologias.com/Catalogos/Manual_Unity.pdf)> [Consulta: 3 de Junio de 2013]



